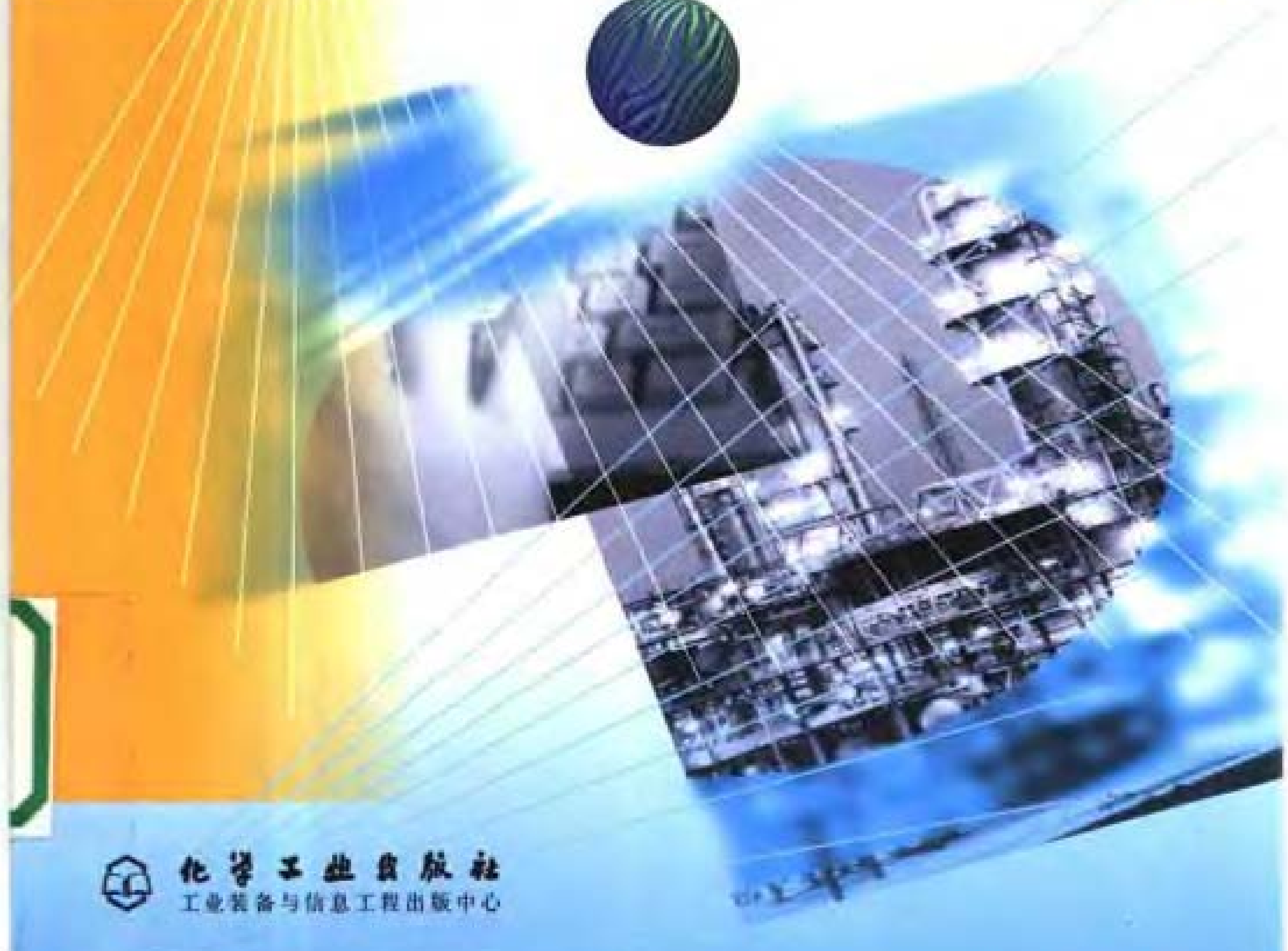
计算机工业应用技术丛书

JISUANJI GONGYE YINGYONG JISHU CONGSHU

数字信号处理及其MATLAB实现

赵红怡　张常年　编著



化学工业出版社
工业装备与信息工程出版中心

计算机工业应用技术丛书

数字信号处理及其MATLAB实现

赵红怡 张常年 编著

化学工业出版社
工业装备与信息工程出版中心
·北 京·

(京)新登字 039 号

图书在版编目(CIP)数据

数字信号处理及其 MATLAB 实现/赵红怡，张常年编著.
北京：化学工业出版社，2002.1
（计算机工业应用技术丛书）
ISBN 7-5025-3471-7

Ⅰ. 数… Ⅱ.①赵…②张… Ⅲ. 数字信号-信号-处理-软件包，MATLAB Ⅳ. TN911.72

中国版本图书馆 CIP 数据核字（2001）第 069996 号

计算机工业应用技术丛书
数字信号处理及其 MATLAB 实现
赵红怡 张常年 编著
责任编辑：李玉晖
责任校对：陶燕华
封面设计：于 兵
*
化学工业出版社
工业装备与信息工程出版中心 出版发行
(北京市朝阳区惠新里 3 号 邮政编码 100029)
发行电话：(010) 64918013
http://www.cip.com.cn
*
新华书店北京发行所经销
北京云浩印刷厂印刷
三河市东柳装订厂装订
开本 787×1092 毫米 1/16 印张 15 字数 363 千字
2002 年 1 月第 1 版 2002 年 1 月北京第 1 次印刷
印 数：1－4000
ISBN 7-5025-3471-7/TP・293
定 价：29.00 元

计算机工业应用技术丛书
编写委员会

序

我国在“十五”期间和今后相当长的时期内将大力发展国民经济和推进社会信息化。这是覆盖现代化建设全局的战略举措。以信息化带动工业化，改造传统产业、发展以信息技术为代表的高新技术产业，从而推进国家现代化建设，已经成为全社会的共识。信息化给企业的经营、管理和发展带来了前所未有的冲击、挑战和机遇，信息化是必然趋势。

当前计算机应用朝着多领域发展，信息化技术涉及多方面的工作，主要包括计算机的广泛利用；企业内部网的建立并与外界实现网络互联；方便访问和利用的信息资源；生产过程控制方面的信息技术应用；计算机辅助设计用于设计新产品；企业生产、流通或服务信息系统有效运转并利用信息网络等手段与外界进行商务往来；建立企业综合管理信息系统等等。随着计算机新技术的不断出现，信息化的内容和工作也将不断扩展。凡是关心国家信息化建设、从事计算机应用开发工作的科技工作者和专业技术人员，都很有必要了解和掌握计算机技术的进步和计算机应用技术的发展。《计算机工业应用技术丛书》就是为以上目的编写的。

《计算机工业应用技术丛书》一套共八册，300 多万字，涉及了当今计算机应用技术的主要领域。其中，《计算机辅助设计与工程分析》和《计算机辅助制造》论述了 CAD/CAM 的主要技术方法并辅以大量的设计制造实例和经验；《工业企业决策支持系统》、《管理信息系统解决方案》和《数据库与工程应用》从不同的角度论述了信息处理技术在企业管理和办公自动化等领域的应用方法、设计技术和如何开发一个以数据库为中心的信息管理系统技术，介绍了多种理论和实用技术；《计算机通信与工业控制》则从企业自动化生产的角度讨论了计算机通信与控制技术的结合并通过先进的背景技术和丰富实例给予说明；《数字信号处理及其 MATLAB 实现》和《图像处理实用技术》则从另外的角度讨论了计算机信息处理技术的发展和变化，用全新的理论和方法研究和处理信息，使信息的表现更丰富多彩、更实用。

《计算机工业应用技术丛书》参考了国际上相关领域的专著和资料，也融会了作者们长期以来的研究成果和心得。对于从事计算机应用工作和关心计算机技术发展的读者，从这套书中可以得到很多启迪和对一些重要问题的解答。它的出版，对推动企事业单位信息技术的发展和应用会产生积极的影响。

《计算机工业应用技术丛书》立足于应用。在内容组织和编排上从理论到实践、由浅入深、图文并茂、通俗易懂。本套书中阐述的解决方案和开发工具是目前先进的和流行的。对于计算机应用技术人员以及从事计算机应用工作的其他专业的科技人员，它都是一套很有益的参考书。

中国科学院院士 赵鹏大

2001 年 6 月于北京

内 容 提 要

数字信号处理作为自动控制的重要环节在各个工业部门都有广泛的应用。计算机用于数字信号处理，由于其本身强大的数据处理能力，显示出比频谱分析仪等传统处理技术更加高效、灵活的特点。本书介绍了用科学计算语言 MATLAB 实现数字信号处理的方法和实践，所有程序均经作者调试运行过。全书内容包括数字信号处理基本理论及实现、数字滤波器设计与实现、一维信号处理的应用及实现、二维信号处理的应用及实现以及 MATLAB 语言中与数字信号处理有关内容的简单介绍。

本书语言简洁，内容精炼，可供应用数字信号处理的教学、科研、开发人员及相关专业学生参考。

目　录

第 1 章　数字信号处理基本理论及实现

随着计算机和信息科学的飞速发展，信号处理逐渐发展成为一门独立的学科，成为信息科学的重要组成部分，在语音处理、雷达、图像处理、通信、生物医学工程等众多领域中得到广泛应用。

1.1　离散信号与系统

在数字信号处理中，所有信号都是离散时间信号——序列，表示为

$$x(n)=\{\cdots,x(-1),x(0),x(1),\cdots\}\qquad -\infty<n<\infty$$

如具体的离散时间信号 $x(n)=\{1,0.5,1,2.5,3,2,1,2,3\}$ $-2\leqslant n\leqslant 6$，用图形表示，如图 1-1 所示，只在 n 为整数时有意义。

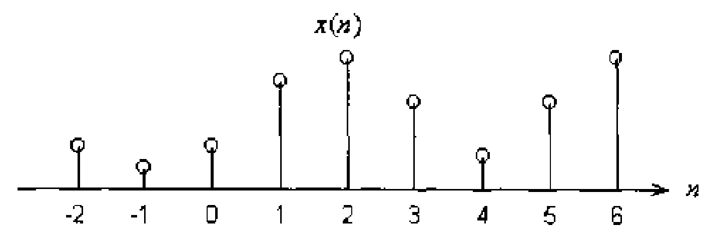


图 1-1　离散时间信号的表示

用 MATLAB 实现如下。

```
%s101.m
n=-2:6;
x=[1,0.5,1,2.5,3,2,0.8,2,3];
stem(n,x);
```

若要表示具有特定采样频率的信号，需定义时间轴向量。例如 $y(t)=\sin(2\pi f_1 t)+2\sin(2\pi f_2 t)$，当 f_1=50Hz, f_2=120Hz, f_s=1000Hz 时的信号为 $y(n)=\sin\left(\dfrac{100\pi}{1000}n\right)+2\sin\left(\dfrac{240\pi}{1000}n\right)$。$y(t)$，$y(n)$ 波形如图 1-2 所示。

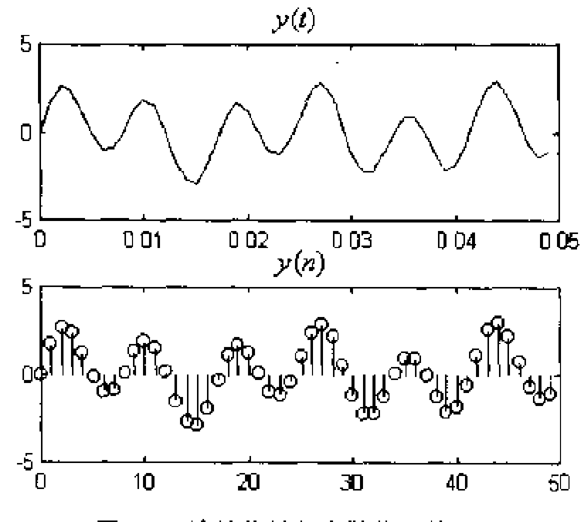


图 1-2　连续信号与离散信号的表示

用 MATLAB 实现如下。

```
%s102.m
f1=50;f2=120;fs=1000;
t=0:1/fs:1;n=t*fs;
y=sin(2*pi*f1*t)+2*sin(2*pi*f2*t);
subplot(211);plot(t(1:50),y(1:50));%画 y(t)的前 0.05s 的值
subplot(212);stem(n(1:50),y(1:50));%画 y(n)前 50 个样点值
```

1.1.1 离散信号的表示与运算

1.1.1.1 信号的表示

（1）典型信号的表示

① 单位脉冲序列 $\delta(n)=\begin{cases}1, & n=0\\0, & n\neq 0\end{cases}$

在MATLAB中，可用函数zeros(1, N)产生一个由N个零组成的列向量，实现有限区间的$\delta(n)$。如

```
x=zeros(1,N);
x(1)=1;
```

如要产生 $\delta(n-n_0)=\begin{cases}1, & n=n_0\\0, & n\neq n_0\end{cases}$，则可以采用MATLAB函数impseq实现如下。

```
function[x,n]=impseq(n0,n1,n2)
n=[n1:n2];x=[(n-n0)==0];
```

② 单位阶跃序列 $u(n)=\begin{cases}1, & n\geqslant 0\\0, & n<0\end{cases}$

可用函数ones(1, N)产生一个由N个1组成的列向量，实现有限区间的$u(n)$。如

```
n=0: N-1;
x=ones(1, N);
```

如要产生 $u(n-n_0)=\begin{cases}1, & n\geqslant n_0\\0, & n<n_0\end{cases}$，则可以采用MATLAB函数stepseq 实现如下。

```
function[x,n]=stepseq(n0,n1,n2)
n=[n1:n2];x=[(n-n0)>=0];
```

③ 矩形序列 $R_N(n)=\begin{cases}1, & N-1\geqslant n\geqslant 0\\0, & 其他\end{cases}$

④ 实指数序列 $x(n)=a^n u(n), a$ 为实数

采用MATLAB 实现如下。

```
n=0:N-1;
x=a.^n;
```

⑤ 正弦序列 $x(n)=\sin(\omega n)$

采用MATLAB 实现如下。

```
n=0:N-1;
x=sin(w*n);
```

⑥ 复指数序列 $x(n)=\mathrm{e}^{(\sigma+\mathrm{j}\omega n)}$

采用MATLAB 实现如下。

```
n=0:N-1;
x=exp((sigema+jw)*n);
```

⑦ 周期序列 $x(n)=x(n+N)$， $-\infty<n<\infty$

x_1是x中的一个周期，要产生3个周期的x序列，采用MATLAB实现如下。

```
x=[x1 x1 x1];
```

（2）常用信号的表示

常用信号的 MATLAB 表示见表 1-1。

表 1-1 常用信号的 MATLAB 函数表示

信号名称	MATLAB 函数	说明	信号名称	MATLAB 函数	说明
三角波或锯齿波	sawtooth（t,width）	width=0.5，产生三角波 width=1，产生锯齿波	非周期三角波 非周期方波	tripuls(t,w,s) rectpuls(t,w)	w 为三角波宽度，s 为斜率 w 为方波宽度
方波	squre(t)		sinc 函数	sinc(t)	

用 MATLAB 实现常用信号的程序举例如下。

```
%s103.m
t=0:0.0001:0.2;
x=sawtooth(2*pi*50*t,1);%锯齿波
subplot(321);plot(t,x);
x=sawtooth(2*pi*50*t,0.5);%三角波
subplot(322);plot(t,x);
x=square(2*pi*50*t);%方波
subplot(323);plot(t,x);axis([0,0.2,-1.5,1.5]);
x=tripuls(t,0.1);%非周期三角波
subplot(324);plot(t,x);axis([0,0.2,-0.1,1.1]);
x=rectpuls(t,0.1);%非周期方波
subplot(325);plot(t,x);axis([0,0.2,-0.1,1.1]);
t=-5:0.1:5;x=sinc(t);
subplot(326);plot(t,x);axis([-5,5,-0.4,1.1]);
```

结果如图 1-3 所示。

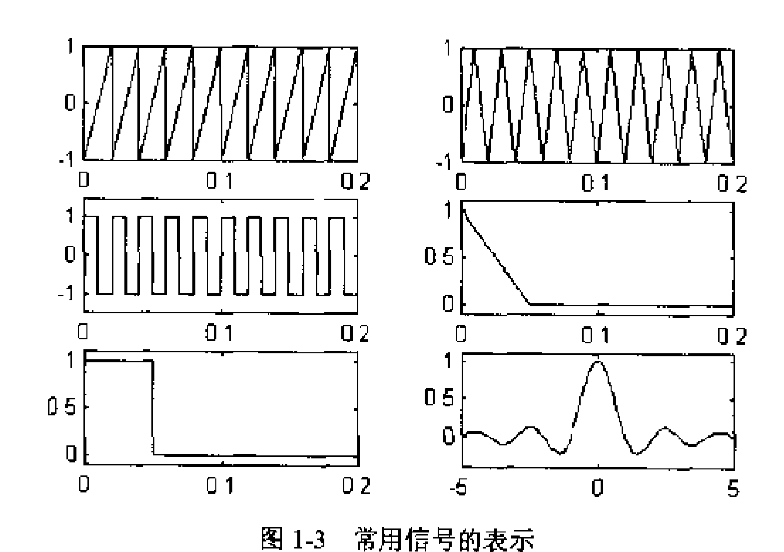


图 1-3 常用信号的表示

1.1.1.2 信号的运算

（1）信号加

$$x(n)=x_1(n)+x_2(n)$$

MATLAB 实现：x=x1+x2

（2）信号乘

$$x(n)=x_1(n)x_2(n)$$

MATLAB 实现：x=x1.*x2

（3）幅度变化

$$y(n)=ax(n)$$

MATLAB 实现：y=a*x

（4）移位

$$y(n)=x(n-m)$$

MATLAB 实现：y=[zeros(1,m) x]

（5）折叠

$$y(n)=x(-n)$$

MATLAB 实现：y=fliplr(x)

（6）采样和

$$y(n)=\sum x(n)$$

MATLAB 实现：y=sum(x(n1:n2))

（7）采样积

$$y(n)=\prod x(n)$$

MATLAB 实现：y=prod(x(n1:n2))

（8）信号能量

$$E_x=\sum_{n=-\infty}^{\infty}|x(n)|^2$$

MATLAB 实现：Ex=sum(abs(x).^2)

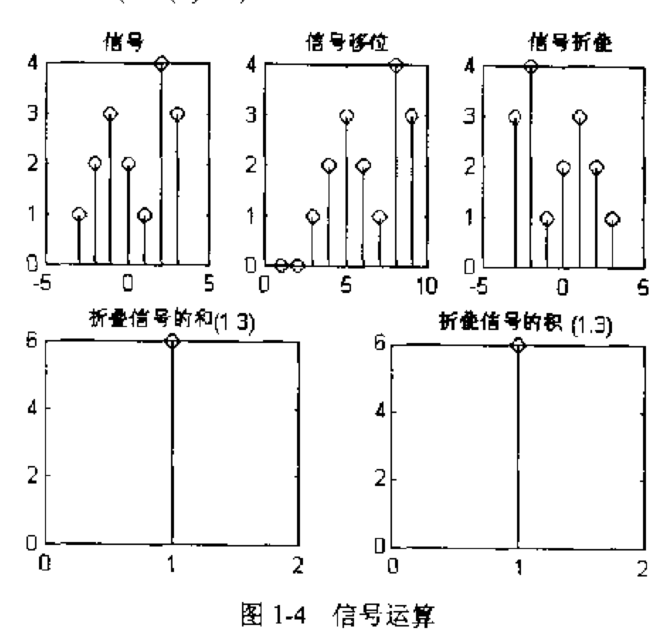


图 1-4 信号运算

用 MATLAB 实现信号基本运算的程序举例如下。

```
%s104.m
n=-3:3;x=[1,2,3,2,1,4,3]; N=length(x);
m=2;n1=1;n2=3;subplot(231);stem(n,x);%表示信号
y=[zeros(1,m) x];subplot(232);stem(y);%信号移位
```

```
y=fliplr(x);subplot(233);stem(n,y);%信号折叠
y=sum(x(n1:n2));subplot(223);stem(y);%信号和
y=prod(x(n1:n2));subplot(224);stem(y);%信号积
Ex=sum(abs(x).^2) %信号能量
Px=sum(abs(x).^2)/N
```

运行结果如图 1-4 所示。

1.1.2 离散系统的表示与实现

一个离散时间系统，输入信号为 $x(n)$，输出信号为 $y(n)$。运算关系用 $T[\ ]$表示，则输出与输入的关系可以表示为 $y(n)=T[x(n)]$。

（1）线性时不变系统（LTI）的表示

线性时不变系统的输入输出关系可通过单位脉冲响应 $h(n)$表示为

$$y(n)=x(n)*h(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)$$

式中*表示卷积运算。

（2）线性时不变系统的实现

可物理实现的线性时不变系统是稳定的、因果的。这种系统的单位脉冲响应是因果的（单边）且绝对可和的，即

$$h(n)=0,\quad n<0$$

$$\sum_{n=-\infty}^{\infty}h(n)<\infty$$

在 MATLAB 中采用函数 conv 实现卷积运算，即

y=conv(x,h);

它默认序列从 n=0 开始。

[例 1] 设某 LTI 的单位脉冲响应 $h(n)=0.8^n u(n)$

（1）判断此系统是否可实现；

（2）当输入为矩形脉冲 $x(n)=u(n)-u(n-10)$ 时，求此 LTI 的输出 $y(n)$；

（3）用 MATLAB 实现，并画出图形。

解：（1）$n<0$ 时 $h(n)$=0，故系统是因果的系统

$\sum_{n=-\infty}^{\infty}h(n)=\sum_{n=0}^{\infty}0.8^n=\dfrac{1}{1-0.8}=5<\infty$，故系统是稳定的系统，所以此系统是可实现的。

（2）$y(n)=x(n)*h(n)$

① $n<0$ 时，$h(n-m)$=0，$0\leqslant m\leqslant 9$

故 $y(n)=0$

② $0\leqslant n\leqslant 9$ 时，$h(n-m)$=1，$0\leqslant m\leqslant n$

故 $y(n)=\sum_{m=0}^{n}1\times 0.8^{(n-m)}=0.8^n\dfrac{1-0.8^{-(n+1)}}{1-0.8^{-1}}=5[1-0.8^{(n+1)}]$

③ $n\geqslant 9$ 时，$h(n-m)$=1，$0\leqslant m\leqslant 9$

故 $y(n)=\sum_{m=0}^{9}1\times 0.8^{(n-m)}=0.8^n\dfrac{1-0.8^{-10}}{1-0.8^{-1}}=5\times(1-0.8^{10})\times 0.8^{(n-9)}$

（3）实现程序如下。

```
%s105.m
x=[ones(1,10)];
x1=[ones(1,10),zeros(1,40)];
N1=length(x);
n1=0:N1-1;
N2=50;n2=0:N2-1;
h=0.8.^n2;
y=conv(x,h);
N=N1+N2-1;n=0:N-1;
subplot(311);stem(n2,x1);
subplot(312);stem(n2,h);
subplot(313);stem(n,y);
```

结果如图 1-5 所示。

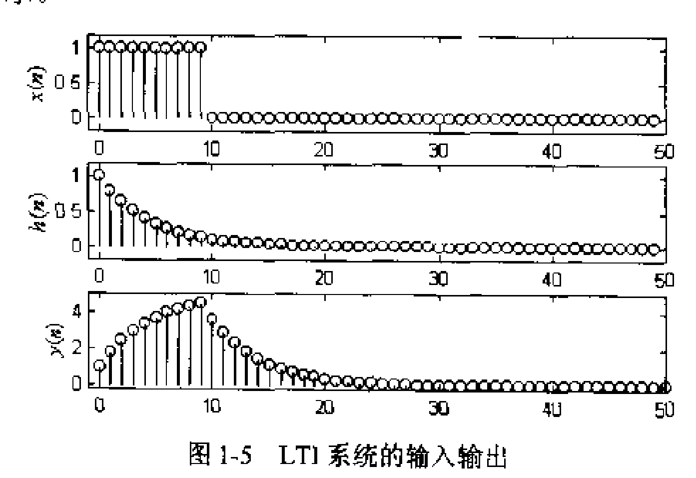


图 1-5 LTI 系统的输入输出

由图 1-5 可见，输入脉冲经过系统后发生了畸变。

如果$x(n)$、$h(n)$的起点不为 0，则可采用 conv_m 计算卷积。

```
function[y,ny]=conv_m(x,nx,h,nh)
%改进卷积程序
nyb=nx(1)+nh(1);
bye=nx(length(x))+nh(length(h));
ny=[nyb,nye];
y=conv(x,h);
```

[例 2] $x(n)=[3,11,7,0,-1,4,2]$，$-3\leqslant n\leqslant 3$；$h(n)=[2,3,0,-5,2,1]$，$-1\leqslant n\leqslant 4$，计算卷积 $y(n)=x(n)*h(n)$。

解：MATLAB 程序实现如下。

```
%s106.m
x=[3,11,7,0,-1,4,2];nx=[-3:3];
h=[2,3,0,-5,2,1];nh=[-1:4];
[y,ny]=conv_m(x,nx,h,nh)
```

运行结果 $y(n)=[6,31,47,6,-51,-5,41,18,-22,-3,8,2]$，$-4\leqslant n\leqslant 7$

1.2 离散时间傅里叶变换与 z 变换

线性时不变系统可以通过线性卷积计算该系统对任意输入的响应，而卷积的含义是，任

何信号可以用单位样本的倍乘和延时的线性组合来表示。把信号表示成为基于复指数信号集$\{e^{j\omega n}\}$的组合，称为离散时间傅里叶变换(DTFT)。在模拟信号和系统中，离散时间傅里叶变换是重要的分析工具。

1.2.1 离散时间傅里叶变换(DTFT)

设序列$x(n)$绝对可和，即$\sum_{n=-\infty}^{\infty}|x(n)|<\infty$，则$x(n)$的离散时间傅里叶变换(DTFT)为

$$X(e^{j\omega})=\sum_{n=-\infty}^{\infty}x(n)e^{-j\omega n} \tag{1-1}$$

$X(e^{j\omega})$的离散时间傅里叶逆变换（IDTFT）为

$$x(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(e^{j\omega})e^{j\omega n}d\omega \tag{1-2}$$

其中ω是数字频率，单位是rad（弧度）。

DTFT的两个重要特性如下。

① 周期性　$X(e^{j\omega})$为连续周期函数，周期为2π，即

$$X(e^{j\omega})=\sum_{n=-\infty}^{\infty}x(n)e^{-j(\omega+2\pi)n}=X(e^{j(\omega+2\pi)})$$

所以分析序列$x(n)$时，只需要得到一个周期的$X(e^{j\omega})$（一般$\omega\in[0,2\pi]$，或者$[-\pi,\pi]$），而不需要研究整个范围$-\infty<\omega<+\infty$。

② 对称性　对于实值的$x(n)$，$X(e^{j\omega})$是共轭对称的，即

$$X(e^{-j\omega})=X^*(e^{j\omega})$$

或者表示成

$$\mathrm{Re}[X(e^{-j\omega})]=\mathrm{Re}[X(e^{j\omega})] \quad \text{（偶对称）}$$
$$\mathrm{Im}[X(e^{-j\omega})]=-\mathrm{Im}[X(e^{j\omega})] \quad \text{（奇对称）}$$
$$|X(e^{-j\omega})|=|X(e^{j\omega})| \quad \text{（偶对称）}$$
$$\angle X(e^{-j\omega})=-\angle X(e^{j\omega}) \quad \text{（奇对称）}$$

所以要画出$X(e^{j\omega})$，只需$X(e^{j\omega})$的半个周期。

［例1］　研究序列$x(n)=0.8^n u(n)$的离散时间傅里叶变换。

解：$x(n)$是绝对可和的，因此它的离散时间傅里叶变换存在

$$X(e^{j\omega})=\sum_{n=-\infty}^{\infty}x(n)e^{-j\omega n}=\frac{e^{j\omega}}{e^{j\omega}-0.8}$$

MATLAB程序实现如下。

```
%s107.m
%[0,2pi]区间分为501点
n=0:50;x=(0.8).^n;%输入序列 x(n)=0.8^nu(n)
subplot(221);stem(n,x); title('输入序列');
w=[0:1:500]*2*pi/500;
X=exp(j*w)./(exp(j*w)-0.8*ones(1,501));%离散时间傅里叶变换
```

```
magX=abs(X);angX=angle(X);
subplot(223);plot(w/pi,magX);
xlabel('以pi为单位的频率');title('离散时间傅里叶变换幅度');
subplot(224);plot(w/pi,angX);
xlabel('以pi为单位的频率');title('离散时间傅里叶变换相位');
```

结果如图1-6所示。

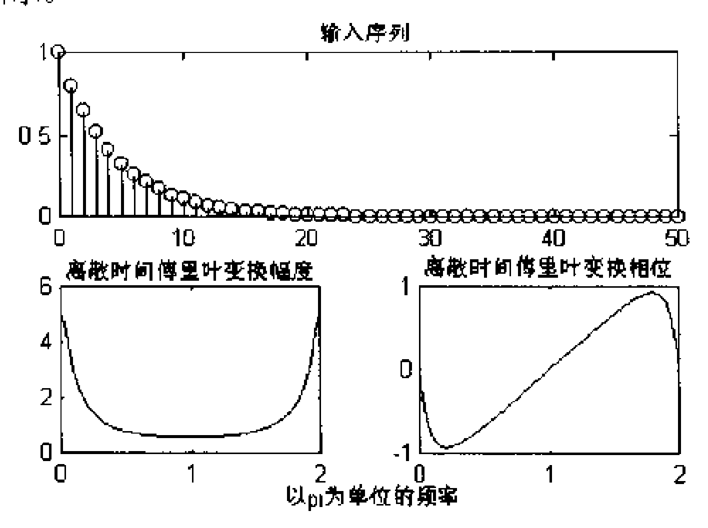


图1-6 离散时间傅里叶变换例1

[例2] 设$x(n)=R_N(n)$，求$x(n)$离散时间傅里叶变换。

解：

$$X(e^{j\omega})=\sum_{n=-\infty}^{\infty}R_N(n)e^{-j\omega n}=\frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}e^{-j(N-1)\frac{\omega}{2}}$$

设$N=7$，其幅度和相位随频率变化曲线如图1-7所示。

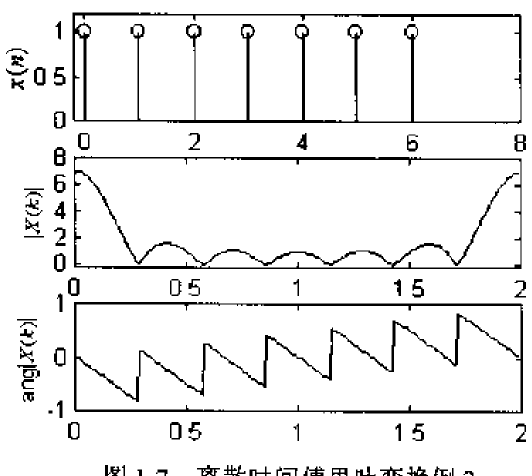


图1-7 离散时间傅里叶变换例2

MATLAB程序实现如下

```
%s108.m
N=7;n=0:N-1;x=[ones(1,N)];
k=0:199;w=(pi/100)*k;%[0, 2pi]轴分为200点
X=x*(exp(-j*pi/100)).^(n'*k);%用矩阵向量乘法求DTFT
```

```
magX=abs(X);angX=angle(X);
subplot(311);stem(n,x);
subplot(312);plot(w/pi,magX);
subplot(313);plot(w/pi,angX/pi);
```

运行结果如图 1-7 所示。

1.2.2 DTFT 的性质

① 线性　离散时间傅里叶变换是一个线性变换，即

设 $X_1(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_1(n)e^{-j\omega n}$； $X_2(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_2(n)e^{-j\omega n}$

那么 $\sum_{n=-\infty}^{\infty}[\alpha x_1(n) + \beta x_2(n)]e^{-j\omega n} = \alpha X_1(e^{j\omega}) + \beta X_2(e^{j\omega})$

② 时移　时域中的移位对应于频域中的相移。

$$\sum_{n=-\infty}^{\infty} x(n-k)e^{-j\omega n} = X(e^{j\omega})e^{-j\omega k}$$

③ 频移　乘以复指数对应于一个频移。

$$\sum_{n=-\infty}^{\infty}[x(n)e^{j\omega_0 n}]e^{-j\omega n} = X(e^{j(\omega-\omega_0)})$$

④ 共轭　时域中的共轭对应于频移中的折叠共轭。

$$\sum_{n=-\infty}^{\infty} x^*(n)e^{-j\omega n} = X^*(e^{-j\omega})$$

⑤ 折叠　时域中的折叠对应于频域中折叠。

$$\sum_{n=-\infty}^{\infty} x(-n)e^{-j\omega n} = X(e^{-j\omega})$$

⑥ 序列卷积　时域中的卷积对应于频域中乘积。

$$\sum_{n=-\infty}^{\infty}[x_1(n) * x_2(n)]e^{-j\omega n} = X_1(e^{-j\omega})X_2(e^{-j\omega})$$

1.2.3 *z* 变换与 *z* 域的系统描述

离散时间傅里叶变换是一种复指数序列表示离散信号的方法，对 LTI 系统可以方便的进行频域分析。但其存在两方面的缺点，其一，在实际应用中许多有用信号，它们的 DTFT 不存在。其二，系统对初始条件的暂态响应无法用 DTFT 方法来计算。

（1）z 变换

为克服上述缺点，把离散时间傅里叶变换方法推广，称为 z 变换。z 变换是对离散时间信号和系统的复频域分析，在数字信号处理中占非常重要的地位。

序列的 z 变换表示为

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{1-3}$$

收敛域表示：$R_{z_-} < |z| < R_{z_+}$

R_{z_-} 和 R_{z_+} 称为收敛半径。

（2）z 域的系统描述

系统函数 $H(z)=\sum_{n=-\infty}^{\infty}h(n)z^{-n}$

$H(z)$的收敛域为 $1\leqslant z\leqslant\infty$。

利用 z 变换的卷积性质，可得 LTI 在 z 域中的表示

$$Y(z)=X(z)H(z) \tag{1-4}$$

以差分方程表示的系统 $y(n)+\sum_{k=1}^{N}a_k y(n-k)=\sum_{m=0}^{M}b_m x(n-m)$

可直接写出 $H(z)=\dfrac{\sum_{m=0}^{M}b_m z^{-m}}{1+\sum_{k=1}^{N}a_k z^{-k}}=b_0 z^{(N-M)}\dfrac{\prod_{m=1}^{M}(z-c_m)}{\prod_{k=1}^{N}(z-d_k)}$

当 $H(z)$的收敛域包含单位圆时，可计算系统的频率响应

$$H(\mathrm{e}^{\mathrm{j}\omega})=b_0\mathrm{e}^{\mathrm{j}(N-M)\omega}\cdot\frac{\prod_{m=1}^{M}(\mathrm{e}^{\mathrm{j}\omega}-c_m)}{\prod_{k=1}^{N}(\mathrm{e}^{\mathrm{j}\omega}-d_k)}$$

式中 $z=c_m$ 是 $H(z)$的零点，$z=d_k$ 是 $H(z)$的极点，它们完全由差分方程的系数决定。

MATLAB 中用 filter 求解差分方程的完全解；freqz 绘出幅频和相频曲线。

[例 3] 一个线性时不变系统，描述它的差分方程为

$$y(n)-0.5y(n-1)=x(n)+0.5x(n-1)$$

（1）求系统的脉冲响应，确定系统稳定性。

（2）如果此系统的输入为 $x(n)=\cos\left(\dfrac{\pi}{10}n+\dfrac{\pi}{4}\right)u(n)$，求系统的响应 $y(n)$。

解：根据差分方程可得

$$H(z)=\frac{1+0.5z^{-1}}{1-0.5z^{-1}}$$

$$h(n)=0.5^n[u(n)+u(n-1)]$$

$$H(\mathrm{e}^{\mathrm{j}\omega})=\frac{1+0.5\mathrm{e}^{-\mathrm{j}\omega}}{1-0.5\mathrm{e}^{-\mathrm{j}\omega}}$$

$$\left|H\left(\frac{\pi}{10}\right)\right|=2.7 \qquad \varphi\left(\frac{\pi}{10}\right)=-\frac{\pi}{8}$$

$$y(n)=x(n)\cdot h(n)=2.7\cos\left(\frac{\pi}{10}n+\frac{\pi}{4}-\frac{\pi}{8}\right)=2.7\cos\left(\frac{\pi}{10}n+\frac{\pi}{8}\right)$$

线性时不变系统，当输入为正弦序列，则输出也为同频正弦序列，其幅度和相位受 $H(\mathrm{e}^{\mathrm{j}\omega})$ 影响。

MATLAB 程序实现如下。

```
%s109.m
b=[1,0.5];a=[1,-0.5];
d=impseq(0,0,30);
n=0:30;%在 0≤n≤30 之间，h(n)截取有限长度
```

```
h=filter(b,a,d);
subplot(231);stem(n,d);title('单位脉冲信号');
subplot(234);stem(n,h);title('单位脉冲响应');
m=0:30;%在 0≤n≤30 之间，x(n)截取有限长度
x=cos(0.2*pi*m+pi/4);
y=filter(b,a,x);
subplot(233);stem(m,x);title('输入信号');
subplot(236);stem(m,y);title('输出信号');
w=[0:500]*2*pi/500;
H=freqz(b,a,w);
M=abs(H);A=angle(H);
subplot(232);plot(w/pi,M);title('幅度响应');
subplot(235);plot(w/pi,A/pi);title('相位响应');
```

运行结果如图 1-8 所示。

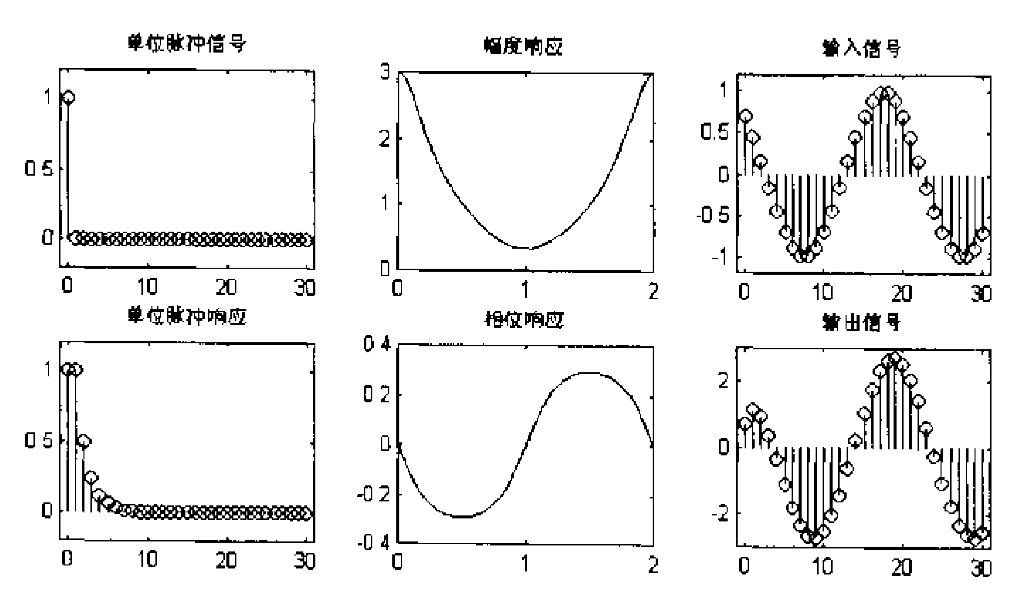


图 1-8　LTI 系统的输入输出和单位脉冲响应

1.2.4　信号的采样与重构

因为数字信号处理技术比模拟信号处理技术有许多优点，所以在工业实际应用中往往把模拟信号变为数字信号，在数字域处理完毕后，若需要再转换为模拟信号。

（1）采样定理及 A/D

令 $x_a(t)$为一绝对可积的模拟信号。它的连续时间傅里叶变换为

$$X_a(j\Omega)=\int_{-\infty}^{+\infty}x_a(t)e^{-j\Omega t}dt$$

式中　Ω——模拟频率，rad/s。

现以间隔 T_s 对 $x_a(t)$进行采样，得到离散时间信号 $x(n)$

$$x(n)=x_a(nT_s)$$

令 $X(e^{j\omega})$为 $x(n)$的离散时间傅里叶变换，可以证明

$$X(e^{j\omega})=\frac{1}{T_s}\sum_{m=-\infty}^{\infty}X_a\left[j\left(\frac{\omega}{T_s}-\frac{2\pi}{T_s}m\right)\right]\tag{1-5}$$

式中　ω ——数字频率，rad。

模拟和数字频率的关系为$\omega = \Omega T_s$，采样频率$f_s=1/T_s$。

式（1-5）的图示见图1-9，由此得出采样定理。

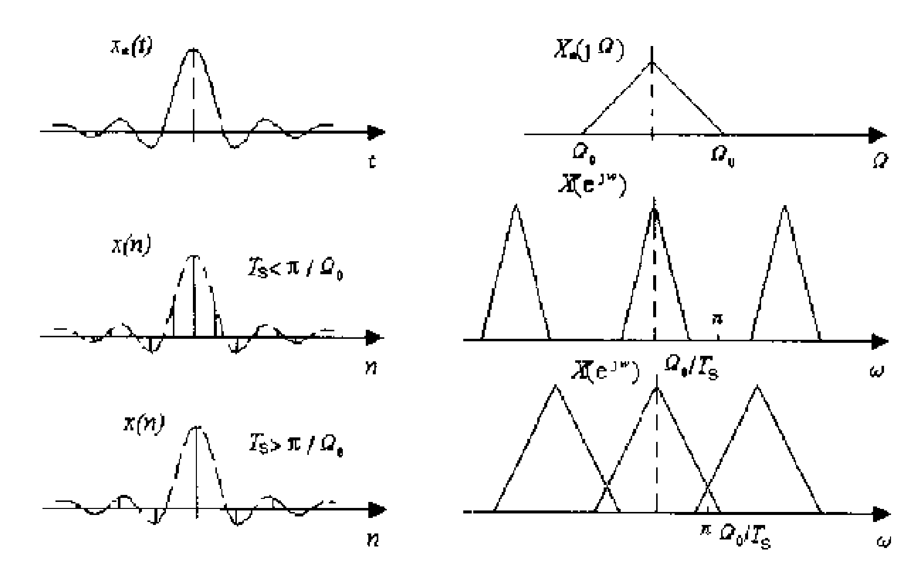


图1-9　采样后频谱周期延拓

采样定理：设$x_a(t)$是带限信号，最高截止频率为f_c。如果采样频率$f_s>2f_c$，则可以唯一地由它的采样值不失真地恢复出原模拟信号；否则会在$x(n)$中产生混叠。$2f_c$称为奈奎斯特频率，$f_s/2$称为折叠频率。

实际完成将模拟信号转换成数字信号的是模数转换器，简称A/D。

［例4］　令$x_a(t)=e^{-1000|t|}$，求出并绘制其傅里叶变换$X_a(j\Omega)$。用三个不同的采样频率对其进行采样，分别求出并画出离散时间傅里叶变换$X(e^{j\omega})$。三个频率分别为：（1）f_s=5kHz；（2）f_s=2kHz；（3）f_s=1kHz。

MATLAB程序实现如下。

```
%s110.m
Dt=0.00005;t=-0.005:Dt:0.005;xa=exp(-1000*abs(t));%模拟信号
Wmax=2*pi*2000;K=500;k=0:1:K;W=k*Wmax/K;
Xa=xa*exp(-j*t'*W)*Dt;Xa=real(Xa);%连续时间傅里叶变换
W=[-fliplr(W),W(2:501)];%频率从-Wmax to Wmax
Xa=[fliplr(Xa),Xa(2:501)];%Xa介于-Wmax和Wmax间
subplot(221);plot(t*1000,xa);xlabel('时间(毫秒)');
ylabel('xa(t)');title('模拟信号')
subplot(222);plot(W/(2*pi*1000),Xa*1000);xlabel('频率(kHz)');
ylabel('Xa(jw)');title('连续时间傅里叶变换')
Ts=0.0002;n=-25:1:25;x=exp(-1000*abs(n*Ts));%离散信号
K=500;k=0:1:K;w=pi*k/K;
X=x*exp(-j*n'*w);X=real(X);%离散时间傅里叶变换
w=[-fliplr(w),w(2:K+1)];
X=[fliplr(X),X(2:K+1)];
subplot(223);stem(n*Ts*1000,x);
xlabel('时间（毫秒）');gtext('Ts=0.2毫秒');
```

```
ylabel('x1(n)');title('离散信号')
subplot(224);plot(w/pi,X);xlabel('频率(弧度)');
ylabel('X1(w)');title('离散时间傅里叶变换')
```

结果如图 1-10 所示，可见 $x_a(t)$的带宽是 2kHz，（1）的采样频率大于 4kHz，所以频谱不产生混叠；（2）、（3）的采样频率小于 4kHz，所以频谱产生混叠。

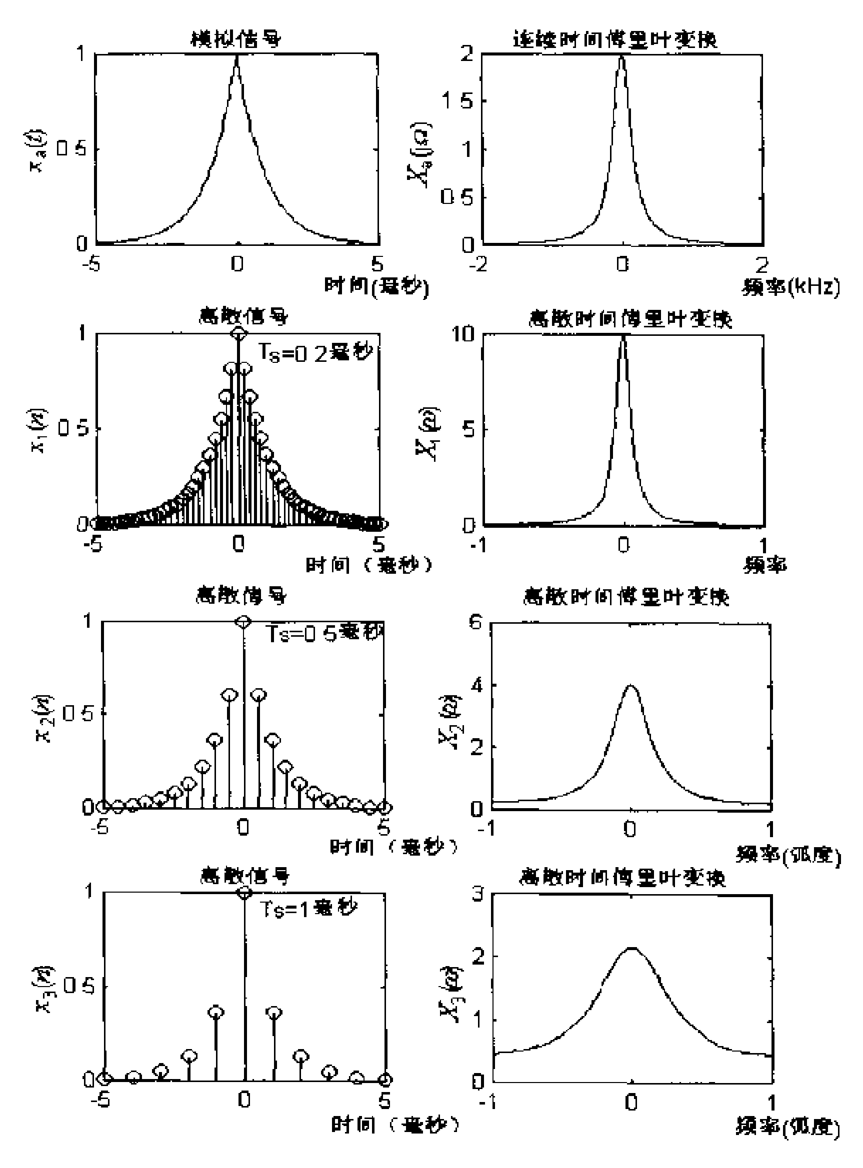


图 1-10　不同采样率下的 DTFT

（2）采样恢复及 D/A

从采样信号重构原信号采用内插公式

$$x_a(t)=\sum_{m=-\infty}^{\infty}x(m)\frac{\sin\left[\frac{\pi}{T_s}(t-mT_s)\right]}{\frac{\pi}{T_s}(t-mT_s)} \tag{1-6}$$

1.3　离散傅里叶变换

离散时间傅里叶变换提供了绝对可和序列在频域中的表示方法，z 变换提供任意序列的

频域表示方法。这两种变换的共同特点是：①适用无限长序列；②变换的结果是连续函数，从计算的角度来看这是不利的。

对有限长序列可采用离散傅里叶变换（简称 DFT），它是可利用计算机进行数值计算的变换，并且存在快速算法，从而使信号的实时处理和设备的简化得以实现。

1.3.1 离散傅里叶变换定义

（1）DFT的定义

设 $x(n)$ 是一个长度为 N 的有限长序列，定义 $x(n)$ 的 N 点离散傅里叶变换为

$$X(k)=\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn} \qquad k=0,1,\cdots,N-1 \tag{1-7}$$

$X(k)$ 的离散傅里叶反变换为

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn} \qquad n=0,1,\cdots,N-1 \tag{1-8}$$

离散傅里叶变换的MATLAB实现如下。

```
function[Xk]=dft(xn,N)
n=[0:1:N-1];%n 的行向量
k=[0:1:N-1];%k 的行向量
WN=exp(-j*2*pi/N);%旋转因子
nk=n'*k;%产生一个含 nk 值的 N 乘 N 维矩阵
WNnk=WN.^nk;%DFT 矩阵
Xk=xn*WNnk;%DFT 系数的行向量
```

离散傅里叶反变换的MATLAB实现如下。

```
function[xn]=idft(Xk,N)
n=[0:1:N-1];%n 的行向量
k=[0:1:N-1];%k 的行向量
WN=exp(-j*2*pi/N);%旋转因子
nk=n'*k;%产生一个含 nk 值的 N 乘 N 维矩阵
WNnk=WN.^(-nk);%IDFT 矩阵
xn=(Xk*WNnk)/N;%IDFT 系数的行向量
```

（2）DFT与 z 变换的关系

设序列 $x(n)$ 的长度为 N，其 z 变换为

$$X(z)=\sum_{n=0}^{N-1}x(n)z^{-n} \tag{1-9}$$

比较式（1-7）、式（1-9）可得

$$X(k)=X(z)\Big|_{z=e^{\mathrm{j}\frac{2\pi}{N}k}} \qquad 0\leqslant k\leqslant N-1 \tag{1-10}$$

或

$$X(k)=X(\mathrm{e}^{\mathrm{j}\omega})\Big|_{\omega=\frac{2\pi}{N}k} \qquad 0\leqslant k\leqslant N-1 \tag{1-11}$$

DFT的物理意义：序列 $x(n)$ 的 N 点 DFT 是 $x(n)$ 的 z 变换在单位圆上的 N 点等间隔采样；$X(k)$ 为 $x(n)$ 的离散时间傅里叶变换 $X(\mathrm{e}^{\mathrm{j}\omega})$ 在区间[0，2π]上的 N 点等间隔采样。

（3）DFT 隐含周期性

在 DFT 变换对中，$x(n)$与 $X(k)$均为有限长序列，设$W_N^{kn}=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn}$，由于$W_N^{kn}$的周期性，使得 $x(n)$与 $X(k)$隐含周期性，且周期为 N。

［例 1］ $x(n)=R_5(n)$，求 N 分别取=10、20 时的 $X(k)$。

解：设 N=10，则

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}=\sum_{n=0}^{9}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{10}kn}=\mathrm{e}^{\mathrm{j}\frac{4\pi}{10}k}\frac{\sin\left(\frac{\pi}{2}k\right)}{\sin\left(\frac{\pi}{10}k\right)}\qquad k=0,1,\cdots,9$$

设 N=20，则

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}=\sum_{n=0}^{19}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{20}kn}=\mathrm{e}^{-\mathrm{j}\frac{4\pi}{20}k}\frac{\sin\left(\frac{\pi}{4}k\right)}{\sin\left(\frac{\pi}{20}k\right)}\qquad k=0,1,\cdots,19$$

```
%s111.m
n=0:4;x=[ones(1,5)];
k=0:999;w=(pi/500)*k;
X=x*(exp(-j*pi/500)).^(n'*k);%计算离散时间傅里叶变换 X(e^jw)
Xe=abs(X);
subplot(321);stem(n,x);ylabel('x(n)');
subplot(322);plot(w/pi,Xe);ylabel('|X(ejw)|');
N=10;x=[ones(1,5),zeros(1,N-5)];
n=0:1:N-1;
X=dft(x,N);%N=10 点离散傅里叶变换
magX=abs(X);phaX=angle(X)*180/pi;
k=(0:length(magX)'-1)*N/length(magX);
subplot(323);stem(n,x);ylabel('x(n)');
subplot(324);stem(k,magX);axis([0,10,0,5]);ylabel('|X(k)|');
N=20;x=[ones(1,5),zeros(1,N-5)];
n=0:1:N-1;
X=dft(x,N);%N=20 点离散傅里叶变换
magX=abs(X);phaX=angle(X)*180/pi;
k=(0:length(magX)'-1)*N/length(magX);
subplot(325);stem(n,x);ylabel('x(n)');
subplot(326);stem(k,magX);axis([0,20,0,5]);ylabel('|X(k)|');
```

结果如图 1-11 所示。

结论：

① 填零是给原始序列填零的运算，会给原始序列的离散时间傅里叶变换提供间隔较密的样本。

② 为画出 $X(\mathrm{e}^{\mathrm{j}\omega})$，只需要 5 点的 $X(k)$用内插公式即可得到 $X(\mathrm{e}^{\mathrm{j}\omega})$。但实际上是用 10 或 20 点的 $X(k)$来填充 $X(\mathrm{e}^{\mathrm{j}\omega})$的值。

③ 填零运算提供了较密的频谱，而没有增加任何新的信息，因此它不能提供高分辨率的频谱。

④ 为得到高分辨率的频谱，需从实验或观察中取得更多的数据。

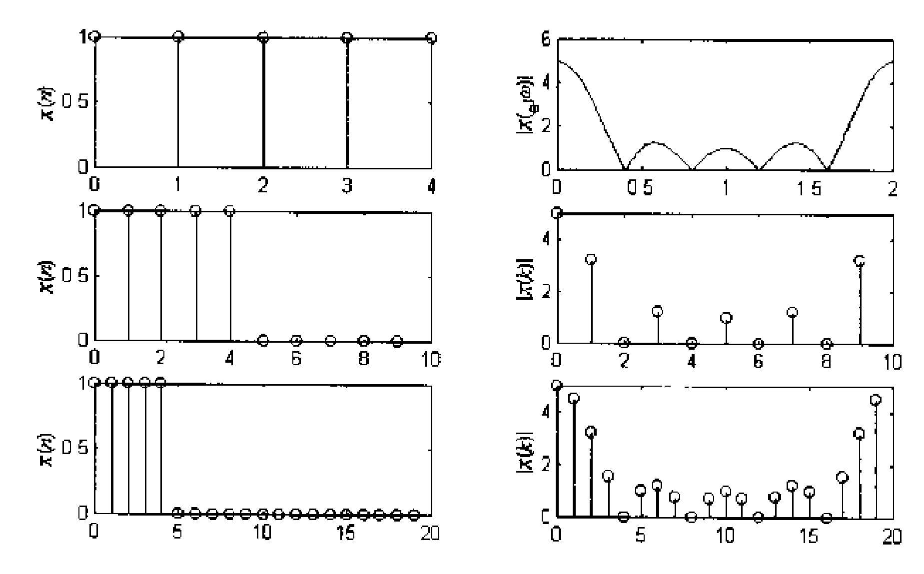


图 1-11　$X(k)$与$X(e^{j\omega})$的关系

[例 2]　为了说明高密度频谱和高分辨率频谱之间的区别，考察序列

$$x(n)=2\cos(0.35\pi n)+\cos(0.5\pi n)$$

① 当 $0\leqslant n<10$ 时，确定并画出 $x(n)$的离散傅里叶变换。

② 当 $x(n)=\begin{cases}x(n) & 0\leqslant n<10\\ 0 & 10\leqslant n<40\end{cases}$ 时，确定并画出 $x(n)$的离散傅里叶变换。

③ 当 $0\leqslant n<40$ 时，确定并画出 $x(n)$的离散傅里叶变换。

```
%s112.m
figure(1);
N=40;M=10;
n=0:N-1;
x=2*cos(0.35*pi*n)+cos(0.5*pi*n);
n1=0:M-1;x1=x(1:M);
subplot(231);stem(n1,x1);title('没有足够采样点的信号');
Y1=dft(x1,M);magY1=abs(Y1);
k1=0:M-1;w1=2*pi/M*k1;
subplot(234);stem(w1/pi,magY1);title('信号的频谱');
n2=0:N-1;x2=[x(1:M) zeros(1,N-M)];
subplot(232);stem(n2,x2);title('填零信号');
Y2=dft(x2,N);magY2=abs(Y2);
k2=0:N-1;w2=2*pi/N*k2;
subplot(235);stem(w2/pi,magY2);title('高密度频谱');
N1=1*N;n=0:N1-1;
x=2*cos(0.35*pi*n)+cos(0.5*pi*n);
subplot(233);stem(n,x);title('有足够采样点的信号');
Y=dft(x,N1);magY=abs(Y);
k=0:N1-1;w=2*pi/N1*k;
subplot(236);stem(w/pi,magY);title('高分辨率频谱');
```

运行后得到如图 1-12 所示的几个图形，结论如下。

① 当 $0 \leqslant n < 10$ 时的序列 $x(n)$与 $X(k)$，从 $X(k)$的图中几乎无法看出有关信号的频谱的信息。

② 将 $x(n)$补 30 个零时的 $x(n)$与 $X(k)$，这时的谱线相当密，但从中很难看出信号的频谱成分，故称为高密度频谱。

③ 将 $x(n)$的长度加长到 40 时的 $x(n)$与 $X(k)$，这时可以清晰的看出信号的频谱成分(ω_1=0.35π，ω_2=0.5π)，故称为高分辨率频谱。

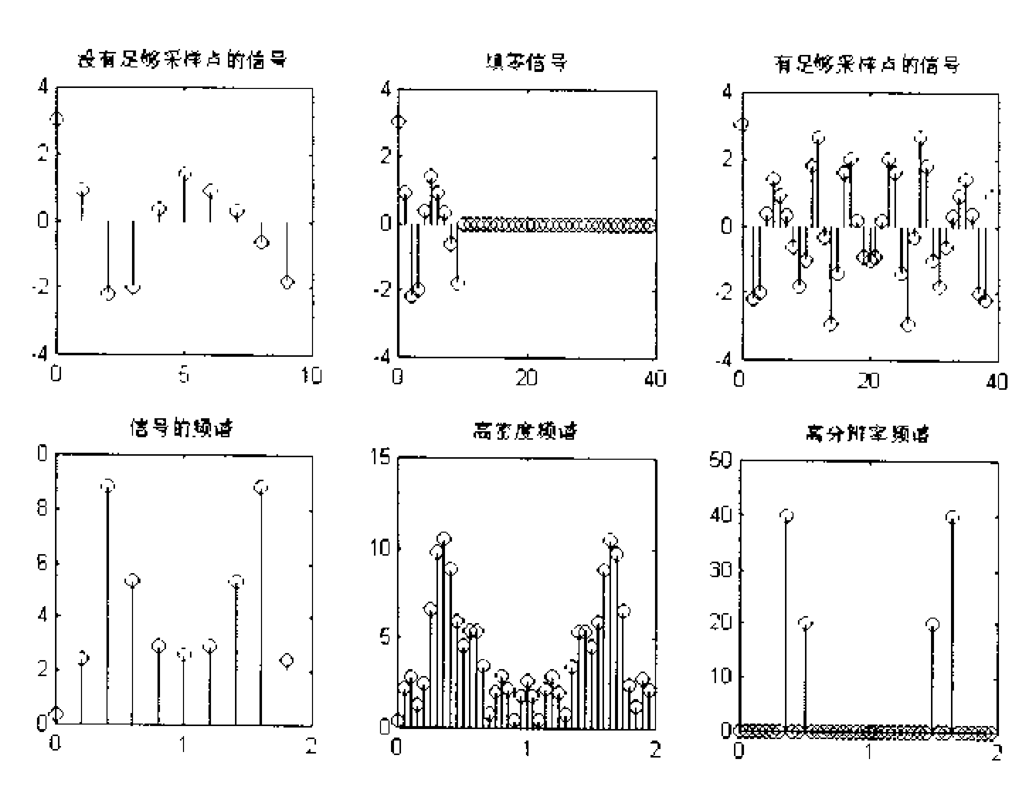


图 1-12 高密度频谱与高分辨率频谱

1.3.2 离散傅里叶变换性质

（1）线性性质

若 $x_1(n)$是 N_1 点，$x_2(n)$是 N_2 点的序列，那么选择 N_3=max[N_1，N_2]，进行 N_3 点 DFT。

$$\mathrm{DFT}[ax_1(n)+bx_2(n)] = a\mathrm{DFT}[x_1(n)] + b\mathrm{DFT}[x_2(n)]$$

（2）圆周移位性质

设 $x(n)$是一个长度为 N 的有限长序列，圆周移位定义为

$$y(n) = x((n+m))_N R_N(n) \tag{1-12}$$

式(1-12)含义：将 $x(n)$以 N 为周期进行周期延拓得到 $\tilde{x}(n) = x((n))_N$，再将 $\tilde{x}(n)$ 左移 m 位得到 $\tilde{x}(n+m)$，最后取 $\tilde{x}(n+m)$ 的主值序列，则得到有限长序列 $x(n)$的圆周移位序列 $y(n)$，$y(n)$仍为长度为 N 的有限长序列，如图 1-13 所示。

$((n))_N$表示 n 对 N 求余，即如果 $n=MN+n_1$，$0 \leqslant n_1 \leqslant N-1$，$M$ 为整数，则

$$((n))_N=n_1$$

圆周移位用 cirshftt 函数实现如下。

```
function y=cirshftt(x,m,N)
if length(x)>N
   error('N 必须>=x 的长度')
end
x=[x zeros(1,N-length(x))];
n=0:N-1;
n=mod(n-m,N);
y=x(n+1);
```

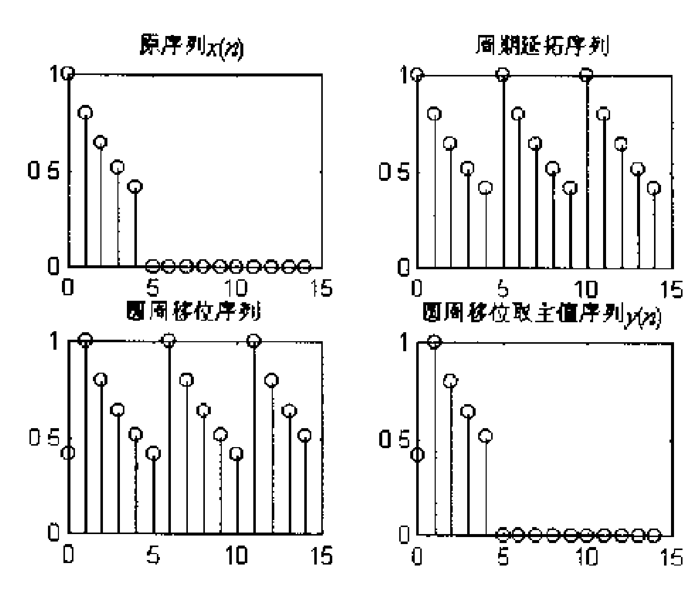


图 1-13 序列的圆周移位过程示意图

[例 3] 序列 $x(n)$={9,8,7,6,5,4,3,2,1}，求分别移 1 位、3 位、5 位、7 位和 9 位的圆周移位。

```
%sl13.m
n=0:8;x=[9,8,7,6,5,4,3,2,1];
y1=cirshftt(x,1,9);
y3=cirshftt(x,3,9);
y5=cirshftt(x,5,9);
y7=cirshftt(x,7,9);
y9=cirshftt(x,9,9);
subplot(611);stem(n,x);ylabel('x(n)');
subplot(612);stem(n,y1);ylabel('y1(n)');
subplot(613);stem(n,y3);ylabel('y3(n)');
subplot(614);stem(n,y5);ylabel('y5(n)');
subplot(615);stem(n,y7);ylabel('y7(n)');
subplot(616);stem(n,y9);ylabel('y9(n)');
```

结果如图 1-14 所示。

（3）圆周卷积定理

如果 $X(k)=X_1(k)\cdot X_2(k)$

则 $x(n)=\text{IDFT}[X(k)]=\sum_{m=0}^{N-1}x_1(m)x_2((n-m))_N R_N(n)$，称为圆周卷积。

简记为 $x(n)=x_1(n)\otimes x_2(n)$

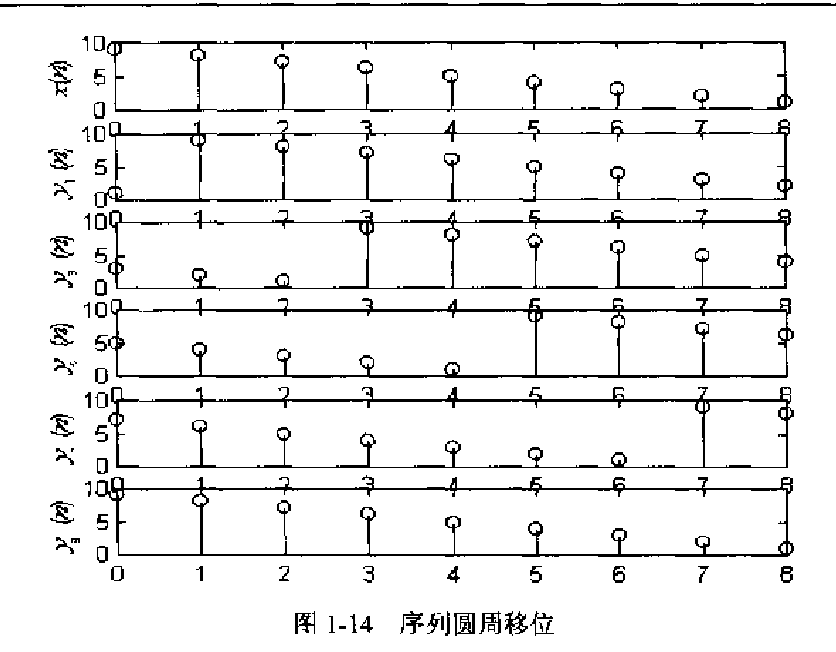


图 1-14 序列圆周移位

[例 4] 计算两序列 $x_1(n)=\{1,2,2,3\}$; $x_2(n)=\{1,2,3,4,3,2\}$的圆周卷积。

```
%s114.m
x1=[1,2,2,3];x2=[1,2,3,4,3,2];
N=length(x1)+length(x2)-1;n=0:N-1;n1=0:N-3;n2=0:N-2;
y1=circonvt(x1,x2,N-2);
y2=circonvt(x1,x2,N-1);y3=circonvt(x1,x2,N);
x1=[x1 zeros(1,N-length(x1))];x2=[x2 zeros(1,N-length(x2))];
X1=dft(x1,N);X2=dft(x2,N);X=X1.*X2;
x=idft(X,N);x=real(x);
subplot(231);stem(n,x1); title('x1(n)');
subplot(232);stem(n,x2); title('x2(n)');
subplot(234);stem(n1,y1); title('N-2 点圆周卷积');
subplot(235);stem(n2,y2); title('N-1 点圆周卷积');
subplot(236);stem(n,y3); title('N 点圆周卷积');
subplot(233);stem(n,x); title('x(n)=IDFT[X(k)]');
```

结果如图 1-15 所示。

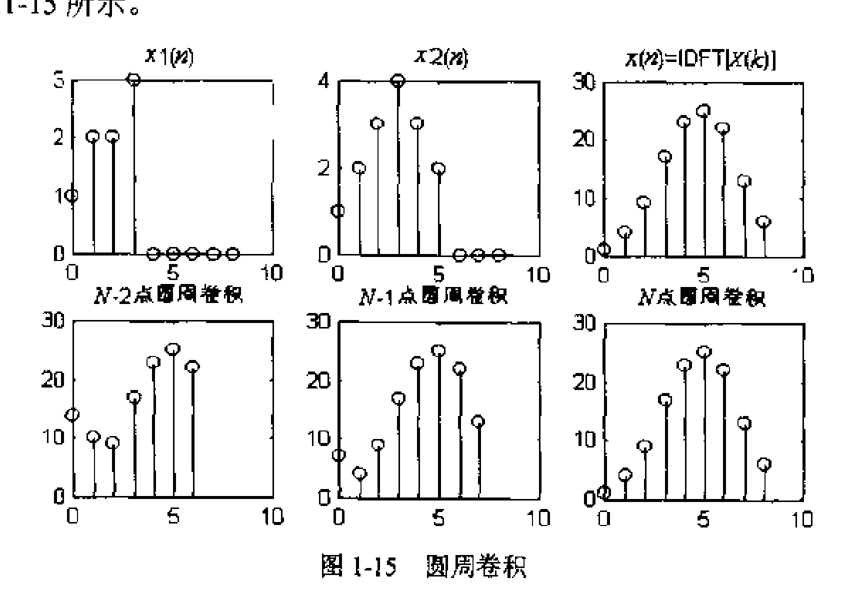


图 1-15 圆周卷积

从此例可见：8 点圆周卷积的第一个样本为 9 点圆周卷积的第一样本和最后一个样本的和，其他点样本相同。7 点圆周卷积的第一个样本为 9 点圆周卷积的第一样本和倒数第二个样本的和，第二个样本为 9 点圆周卷积的第二样本和最后一个样本的和，其他点样本相同。

（4）复共轭序列的 DFT

设 $x^*(n)$是 $x(n)$的复共轭序列，长度为 N

$$X(k)=\text{DFT}[x(n)]$$

则 $$\text{DFT}[x^*(n)]=X^*(N-k),\ 0\leqslant k\leqslant n-1$$

且 $$X(N)=X(0)$$

[例 5] 计算共轭序列 $x(n)=\{1-j,2+2j,3-3j,-4+4j,5-5j\}$的 DFT 和 $x^*(n)$的 DFT。

```
%s115.m
xn=[1-j,2+2j,3-3j,-4+4j,5-5j];
Xk=dft(x,5);x1=(xn').';X=dft(x1,5);n=0:4;
subplot(221);stem(n,abs(xn));title('|x(n)|');
subplot(222);stem(n,abs(Xk));title('|X(k)|');
subplot(223);stem(n,abs(x1));title('|x*(n)|');
subplot(224);stem(n,abs(X));title('|X*(N-k)|');
```

运行结果如图 1-16 所示。

Xk=[7−3i,6.5150−3.9974i,5.7998+10.1568i, −15.7440−6.6847i,1.4292−1.4747i]

X1=[1+i,2−2i,3+3i, 4−4i,5+5i]

X=[7+3i,1.4292+1.4747i, −15.7440+6.6847i,5.7998−10.1568i,6.5150+3.9974i]

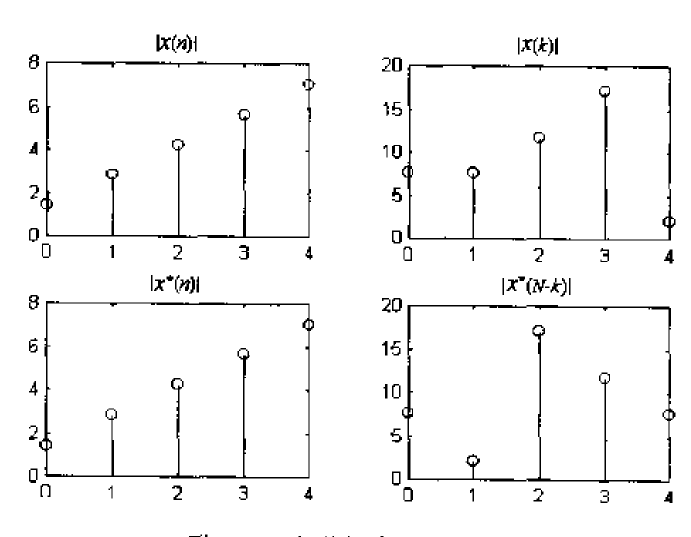


图 1-16 复共轭序列的 DFT

（5）DFT 的共轭对称性

DFT 中的 $x(n)$、$X(k)$均为有限长序列，其对称性是指关于 $N/2$ 点的对称性。

① 有限长共轭对称序列和共轭反对称序列　有限长共轭对称序列用 $x_{ep}(n)$表示，共轭反对称序列用 $x_{op}(n)$表示，则有

$$x_{ep}(n)=x_{ep}^*(N-n),\quad 0\leqslant n\leqslant N-1$$

$$x_{op}(n)=-x_{op}^*(N-n),\quad 0\leqslant n\leqslant N-1$$

任何有限长序列都可分解成共轭对称分量和共轭反对称分量之和，即

$$x(n)=x_{ep}(n)+x_{op}(n) \quad 0\leqslant n\leqslant N-1$$

则

$$x_{ep}(n)=\frac{1}{2}[x(n)+x^*(N-n)] \quad 0\leqslant n\leqslant N-1$$

$$x_{op}(n)=\frac{1}{2}[x(n)-x^*(N-n)] \quad 0\leqslant n\leqslant N-1$$

② DFT 的共轭对称性

如果 $x(n)=x_r(n)+jx_i(n)$，则 $X(k)=X_{ep}(k)+X_{op}(k)$

其中 $\mathrm{DFT}[x_r(n)]=X_{ep}(k)$，$\mathrm{DFT}[jx_i(n)]=X_{op}(k)$

如果 $x(n)=x_{ep}(n)+x_{op}(n)$，则 $X(k)=X_R(k)+jX_I(k)$

其中 $\mathrm{DFT}[x_{ep}(n)]=X_R(k)$，$\mathrm{DFT}[x_{op}(n)]=jX_I(k)$

DFT 的共轭对称性：如果 $x(n)$的 DFT 为 $X(k)$，则 $x(n)$的实部和虚部（包括 j）的 DFT 分别为 $X(k)$的共轭对称分量和共轭反对称分量；而 $x(n)$的共轭对称分量和共轭反对称分量的 DFT 分别为 $X(k)$的实部和虚部乘以 j。

实际应用中利用上述对称性质，可减少 DFT 运算量，提高运算效率。

1.3.3 用 DFT 分析系统

DFT 在数字通信、语音信号处理、图像处理、雷达理论、光学、医学、地震等各个领域得到广泛应用。这些应用一般都是以卷积运算为依据或以 DFT 作为连续时间傅里叶变换的近似为基础。

（1）用 DFT 计算线性卷积

设 $x(n)$为 N 点，$h(n)$为 M 点，则线性卷积 $y_l(n)$为 $N+M-1$ 点序列，即

$$y_l(n)=x(n)*h(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=\sum_{m=0}^{M-1}x(m)h(n-m)$$

如果取 $L=N+M-1$，圆周卷积 $y_c(n)$为

$$y_c(n)=x(n)\otimes h(n)=\sum_{m=0}^{L-1}x(m)h((n-m))_N R_N(n)=\sum_{m=-\infty}^{\infty}y_l(n+qL)\cdot R_N(n)$$

则

$$y_c(n)=y_l(n) \quad 0\leqslant n\leqslant L-1$$

说明可采用 DFT 计算线性卷积，称其为快速卷积，计算框图如图 1-17 所示。

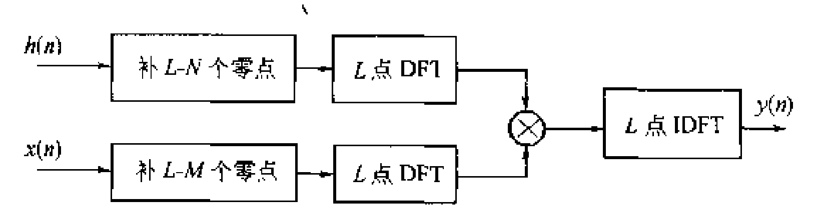


图 1-17 用 DFT 计算线性卷积

在实际应用中经常遇到两个序列的长度相差很大的情况，用上述快速卷积计算线性卷积，要求对短序列补很多零，长序列必须全部输入后才能进行快速计算。因此要求存贮容量大，运算时间长，并行处理延时很大，很难实时处理。而在某些场合，序列长度不定或人为无限长，直接套用上述方法是不行的。解决问题的方法是将长序列分段计算。

重叠保留法：在每段的前端保留原来的输入序列值，用 DFT 实现圆周卷积。

设 $x(n)$中的任意一段长为 N 的序列 $x_i(n)$与长为 M 的 $h(n)$的 N 点圆周卷积为

$$y_i'(n)=x_i(n)\otimes h(n)=\sum_{m=0}^{N-1}x_i(m)h((n-m))_N R_N(n)$$

在 $M-1\leqslant n\leqslant N-1$ 范围内，圆周卷积存与线性卷积的结果一样，则前 $M-1$ 个值应去掉。为了不造成输出信号的遗漏，对 $x(n)$分段时，就需使相邻两段有 $M-1$ 个点的重叠。第一段 $x(n)$由于没有前一段保留信号，则在其前补充 $M-1$ 个零点值。

定义

$$x_k(n)=\begin{cases}x[n+k(N-(M-1))-M+1] & 0\leqslant n\leqslant N-1\\ 0 & 其他\end{cases}$$

则

$$y(n)=\sum_{k=0}^{\infty}y_k(n-k(N-M+1))$$

式中 n——总输出序列 $y(n)$的序号。

$$y_k(n)=\begin{cases}y_k'(n) & M-1\leqslant n\leqslant N-1\\ 0 & 其他\end{cases}$$

［例 6］ 设 $x(n)$={10,9,8,7,6,5,4,3,2,1}，$h(n)$={1,1,–1}按 N=6 用重叠保留方法计算

$$y(n)=x(n)*h(n)$$

解：M=3, N=6，因此分块时每块长度为 6，重复 M–1=2 个取样点。

分块后

$$x_1(n)=\{0,0,10,9,8,7\}$$
$$x_2(n)=\{8,7,6,5,4,3\}$$
$$x_3(n)=\{4,3,2,1,0,0\}$$

这样可分别计算

$$y_1(n)=x_1(n)\otimes h(n)=\{-1,-7,10,19,7,6\}$$
$$y_2(n)=x_2(n)\otimes h(n)=\{7,12,5,4,3,2\}$$
$$y_3(n)=x_3(n)\otimes h(n)=\{4,7,1,0,-1,-1\}$$

舍弃前两个取样值，组合得到 $y(n)$

$$y(n)=\{10,19,7,6,5,4,3,2,1,0,-1,-1\}$$

这与线性卷积的结果一致。

用 MATLAB 实现如下。

```
%s117.m
x=[10,9,8,7,6,5,4,3,2,1];h=[1,1,-1];
x1=[0,0,10,9,8,7];x2=[8,7,6,5,4,3];x3=[4,3,2,1,0,0];
y1=circonvt(x1,h,6);y2=circonvt(x2,h,6);
y3=circonvt(x3,h,6);y=ovrlpsav(x,h,6);
n=0:5;N=length(x)+length(h)-1;n1=0:N-1;n2=0:9;
subplot(241);stem(n,x1);title('x1');axis([0,6,0,10]);
subplot(245);stem(n,y1);title('y1');axis([0,6,-10,20]);
subplot(242);stem(n,x2);title('x2');axis([0,6,0,10]);
subplot(246);stem(n,y2);title('y2');axis([0,6,-10,20]);
```

```
subplot(243);stem(n,x3);title('x3');axis([0,6,0,10]);
subplot(247);stem(n,y3);title('y3');axis([0,6,-10,20]);
subplot(244);stem(n2,x);title('x');axis([0,11,0,10]);
subplot(248);stem(n1,y);title('y');axis([0,11,-10,20]);
```

运行结果如图 1-18 所示。

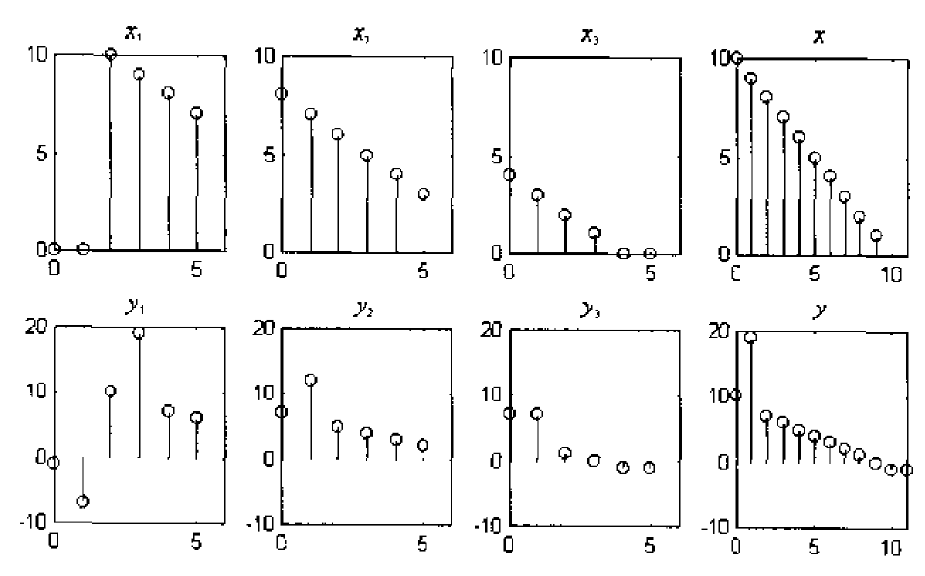


图 1-18 重叠保留方法计算线性卷积

用 ovrlpsav 实现重叠保留法。

```
function[y]=ovrlpsav(x,h,N)
Lenx=length(x);M=length(h);%x 输入序列，h 脉冲响应
M1=M-1;L=N-M1;%N 段长
h=[h zeros(1,N-M)];
x=[zeros(1,M1),x,zeros(1,N-1)];%预置 M-1 个零
K=floor((Lenx+M1-1)/(L));%段数
Y=zeros(K+1,N);
for k=0:K %各段圆卷积
   xk=x(k*L+1:k*L+N);
   Y(k+1,:)=circonvt(xk,h,N);
end
Y=Y(:,M:N)';%去掉前 M-1 个值
y=(Y(:))';%装成输出
```

（2）用 DFT 进行谱分析

信号的谱分析就是计算信号的傅里叶变换。连续信号与系统的傅里叶变换不便于直接用计算机进行计算，其应用受到限制。DFT 是离散信号与系统分析的有力工具，对连续信号与系统可以通过时域采样，应用 DFT 进行近似谱分析。

① 用 DFT 对连续信号进行谱分析 工程实际应用中，经常遇到的连续信号 $x_a(t)$，其频谱 $X_a(j\Omega)$也是连续函数。为了利用 DFT 对 $x_a(t)$进行频谱分析，先对 $x_a(t)$进行时域采样，得到$x(n)=x_a(nT)$，再对 $x(n)$进行 DFT，得到的 $X(k)$则是 $x(n)$的傅里叶变换 $X(e^{j\omega})$在频域区间[0, 2π]上的 N 点等间隔采样。$x(n)$和 $X(k)$均为有限长序列。

由傅里叶变换理论知道，若信号持续时间有限长，则其频谱无限宽；若信号的频谱有限

宽，则其持续时间无限长。所以严格地讲，持续时间有限的带限信号是不存在的。上述两种情况是不满足 DFT 变换条件的。根据采样定理，为了减小采样后产生的频谱混叠失真，可用预滤波法滤除幅度较小的高频成分。为避免采样点数太多导致无法存贮和计算，只好截取有限点进行 DFT。

用 DFT 对连续信号进行谱分析必然是近似的，其近似的结果与信号带宽、采样速率和截取长度有关。从工程的角度看，滤除幅度较小的高频成分和截去幅度很小的部分时间信号是允许的。

[例 7]　对模拟信号 $x_a(t)=2\sin(4\pi t)+5\cos(8\pi t)$以间隔 T 对其采样，得到 N 点序列 $x(n)$，用 N 点 DFT 得到对 $x_a(t)$幅度谱的估计。

（1）T=0.01s，N=40 或 N=50，一个能提供精确的 $x_a(t)$的幅度谱，画出 DFT 的幅度谱。

（2）T=0.005s，N=40 或 N=50，画出 DFT 的幅度谱。

```
%s118.m
T=0.01;N=40;n=0:N-1;t=n*T;
xn=2*sin(4*pi*t)+5*cos(8*pi*t);
Xk=dft(xn,N);
magXk=abs(Xk);
k=(0:length(magXk)'-1)*N/length(magXk);
subplot(241);plot(t,xn);axis([0,0.4,-7.5,7]);
title('T=0.01s,t=0.4s');ylabel('x(t)');
subplot(245);stem(k,magXk);title('T=0.01s,N=40');ylabel('X(k)');
```

结果如图 1-19 所示。

结果说明：采样间隔 T =0.01s，采样点数 N=50 时的幅度谱最精确地 $x_a(t)$的幅度谱，而其他情况都存在频谱泄漏。

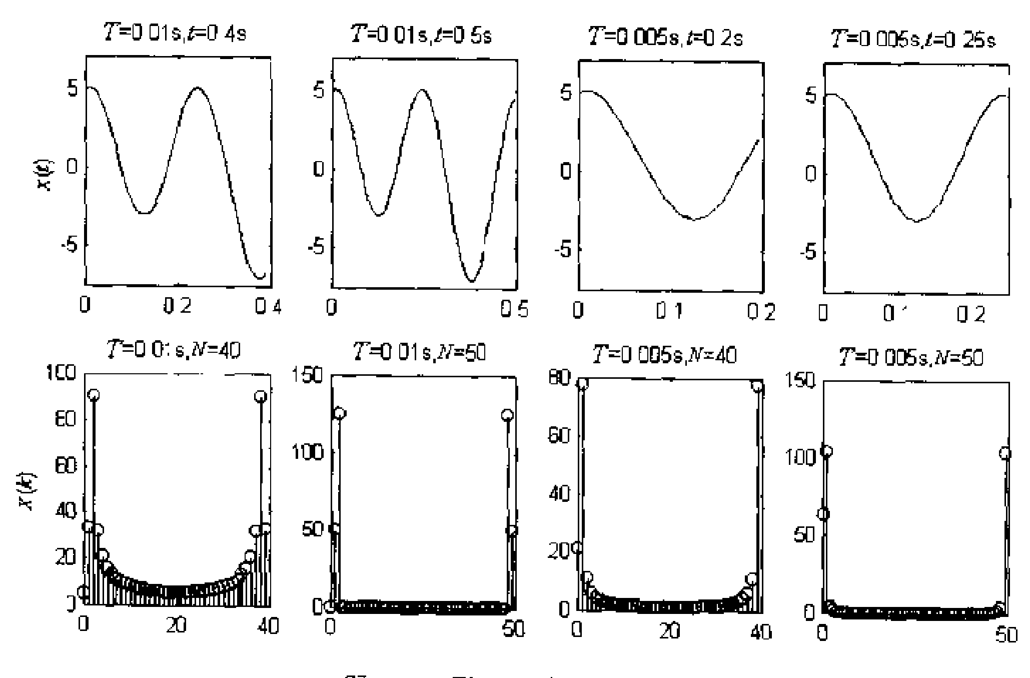


图 1-19　用 DFT 进行频谱分析

[例 8]　已知一模拟信号 $x_a(t)=e^{-t}u(t)$，现以采样率 f_s=20Hz 进行采样。用 DFT 计算当序列长度①L=100，②L=20 时，N=200 点的幅度频谱样值并通过作图与理论上准确的频谱样值

进行比较。

解：原信号的傅里叶变换

$$X_a(j\Omega)=\int_{-\infty}^{\infty}x_a(t)e^{-j\Omega t}dt=\frac{1}{1+j\Omega}$$

其幅度为$|X_a(j\Omega)|=\frac{1}{1+\Omega^2}$

理论上准确的频谱如图 1-20 所示。

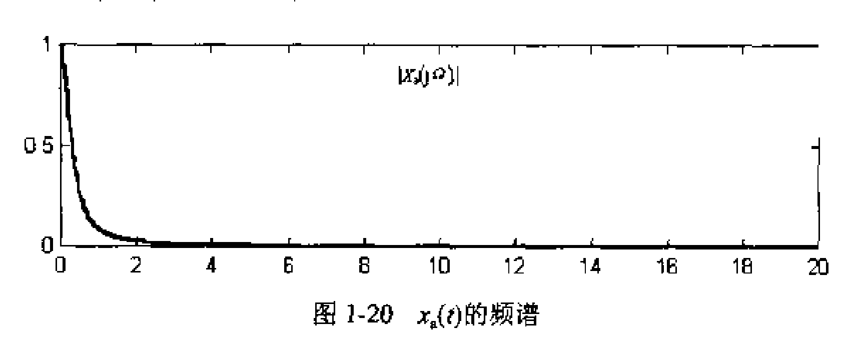


图 1-20 $x_a(t)$的频谱

用 MATLAB 实现如下。

```
%sli9.m
fs=20;
L=100;N=200;n=0:L-1;t1=n/fs;
xn1=exp(-t1);xn=[xn1,zeros(1,N-L)];
Xk1=dft(xn,N);magXk1=abs(Xk1);
k1=(0:length(magXk1)'-1)*N/length(magXk1);
L=20;N=200;n=0:L-1;t2=n/fs;
xn2=exp(-t2);xn=[xn2,zeros(1,N-L)];
Xk2=dft(xn,N);magXk2=abs(Xk2);
k2=(0:length(magXk2)'-1)*N/length(magXk2);
figure(1);
subplot(211);plot(t1,xn1);title('xa(t)  t=5s');
subplot(212);plot(t2,xn2);title('xa(t)  t=1s');
figure(2);
subplot(211); plot(k1,magXk1);title('X(k)  L=100  N=200');
subplot(212); plot(k2,magXk2);title('X(k)  L=20  N=200');
figure(3);
Omeger=0:0.1:20*pi;
Xa=1./(1+Omeger.^2);
subplot(111);plot(Omeger/pi,Xa);title('|Xa(j\Omega)|');
```

计算结果如图 1-21 所示，结论如下。

● 当序列长度为 100，进行 200 点 DFT 计算的结果混叠与泄漏的影响比较小，基本上接近原信号的频谱。因为按给定的 f_s=20Hz，相当于取信号的最高频率 f_h=10Hz，故在[0, f_h]频率范围内的信号能量为

$$E_h=\frac{1}{2\pi}\int_{-20\pi}^{20\pi}|X(j\Omega)|^2d\Omega=0.495$$

而信号的总能量为

$$E_x = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(\mathrm{j}\Omega)|^2 \,\mathrm{d}\Omega = \frac{1}{2\pi}\arctan\Omega \Big|_{-\infty}^{\infty} = 0.5$$

所以 E_h/E_x=99%，基本上满足频谱不混叠的要求。

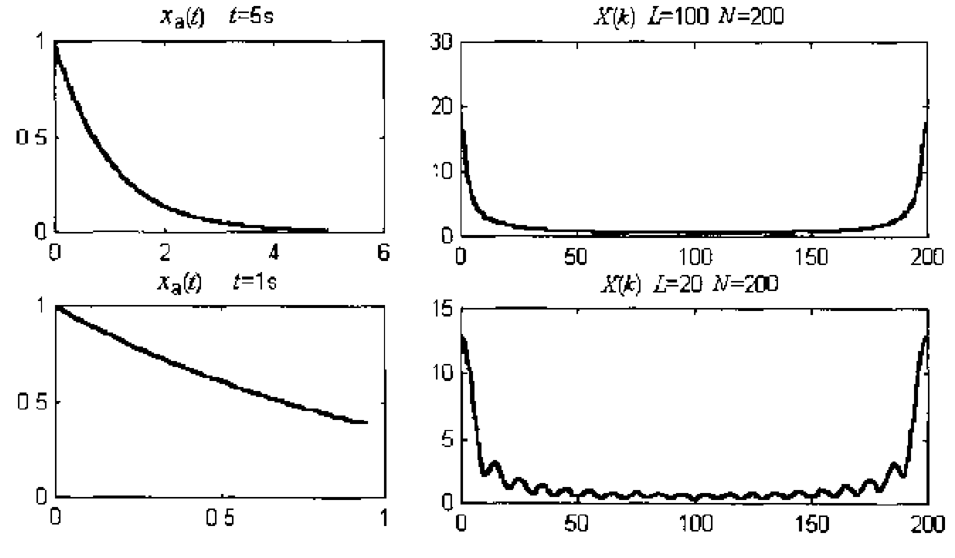


图 1-21　用 DFT 计算的频谱

● 当序列长度为 20，进行 200 点 DFT 计算，由于截取 $x(n)$长度太短

$$x(t)|_{t=LT} = \mathrm{e}^{-LT} = \frac{1}{\mathrm{e}} = 0.3079 \gg 0$$

所以频谱因泄漏出现较大的波动，以致与原信号频谱有较大差别。

② 用 DFT 对离散信号进行谱分析　序列 $x(n)$的在单位圆上的 z 变换就是其傅里叶变换 $X(\mathrm{e}^{\mathrm{j}\omega})$，即

$$X(\mathrm{e}^{\mathrm{j}\omega}) = X(z)|_{z=\mathrm{e}^{\mathrm{j}\omega}}$$

对序列 $x(n)$进行 N 点 DFT 得到 $X(k)$，$X(k)$是 $X(\mathrm{e}^{\mathrm{j}\omega})$在区间$[0,2\pi]$上的 N 点等间隔采样，因此序列的傅里叶变换可利用 DFT 来计算。

实际工作中往往要把信号的观察时间限制在一定的时间间隔内，就需要取用有限个数据，即将信号截断，相当于将信号乘以窗函数。在时域的截断，相当于所研究的频谱与窗函数频谱的周期卷积，周期卷积的结果造成频谱从其正常频谱扩展开来，称为“泄漏”。泄漏效应是 DFT 所固有的，可用加窗函数进行抑制。

加窗函数后，信号成为

$$x_{\mathrm{w}}(n) = x(n)w(n)$$

频谱为

$$X_{\mathrm{w}}(\mathrm{e}^{\mathrm{j}\omega}) = X(\mathrm{e}^{\mathrm{j}\omega}) * W(\mathrm{e}^{\mathrm{j}\omega})$$

频谱的等间隔取样值为

$$X_{\mathrm{w}}(k) = X(k) * W(k)$$

用窗函数加权使有限长度的输入信号周期延拓后，在边界上尽量减少不连续程度，从而达到抑制频谱泄漏的目的。

在实际应用中常用单边表示的窗函数，即窗函数

$$w(n) \quad n = 0,1,\cdots,N-1$$

这种将加权序列向右移动 $N/2$ 点，只影响相位特性，并不影响振幅特性。

矩形窗函数

$$w(n)=1,\quad n=0,1,\cdots N-1$$

三角窗函数

$$w(n)=\begin{cases}\dfrac{n}{N/2} & n=0,1,\cdots,\dfrac{N}{2}\\ w(N-n) & n=\dfrac{N}{2},\dfrac{N}{2}+1,\cdots,N-1\end{cases}$$

汉宁窗函数

$$w(n)=\frac{1}{2}\left[1-\cos\left(\frac{2n}{N}\pi\right)\right]\qquad n=0,1,\cdots,N-1$$

海明窗函数

$$w(n)=0.54-0.46\cos\left(\frac{2n}{N}\pi\right)\qquad n=0,1,\cdots,N-1$$

布来克曼窗函数

$$w(n)=0.42-0.5\cos\left(\frac{2n}{N}\pi\right)+0.08\cos\left(\frac{4n}{N}\pi\right)\qquad n=0,1,\cdots,N-1$$

常用的窗函数特性见表 1-2。

表 1-2　常用的窗函数特性

窗函数	MATLAB 实现	近似过渡过程	精确过渡过程	最小阻带衰减
矩形窗	Boxcar(N)	$4\pi/N$	$1.8\pi/N$	21dB
三角窗	Triang(N)	$8\pi/N$	$6.1\pi/N$	25dB
汉宁窗	Hanning(N)	$8\pi/N$	$6.2\pi/N$	44dB
海明窗	Hamming(N)	$8\pi/N$	$6.6\pi/N$	53dB
布来克曼窗	Blackman(N)	$12\pi/N$	$11\pi/N$	74dB

[例 9]　信号 $x(n)=e^{-0.05n}$，用加窗函数的 DFT 分析其频谱。

```
%s120.m
figure(1);
L=20;N=200;n=0:L-1;
xn=exp(-0.05*n);xn1=[xn,zeros(1,N-L)];
Xk=dft(xn1,N);magXk=abs(Xk);
k=(0:length(magXk)'-1)*N/length(magXk);
subplot(211);stem(k,xn1);title('矩形窗 x(n)');
subplot(212);stem(k,magXk);title('矩形窗 X(K)');
figure(2);
w=(hanning(L))';xn1=xn.*w;
xn2=[xn1,zeros(1,N-L)];
subplot(211);stem(k,xn2);title('汉宁窗 x(n)');
Xk=dft(xn2,N);magXk=abs(Xk);
subplot(212);stem(k,magXk);title('汉宁窗 X(K)');
figure(3);
w=(triang(L))';xn1=xn.*w;
xn2=[xn1,zeros(1,N-L)];
subplot(211);stem(k,xn2);title('三角窗 x(n)');
Xk=dft(xn2,N);magXk=abs(Xk);
subplot(212);stem(k,magXk);title('三角窗 X(K)');
```

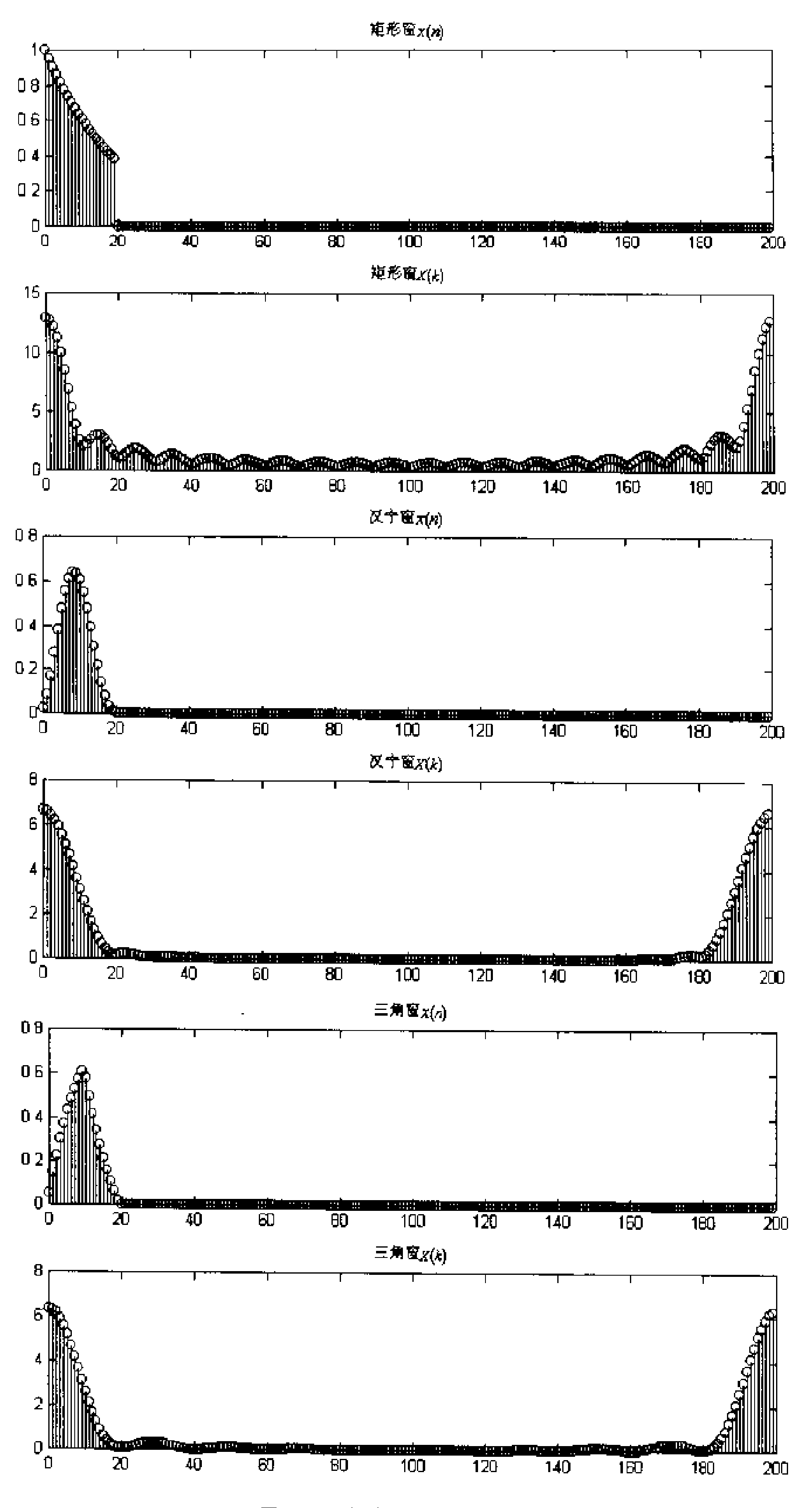


图 1-22　加窗后的信号及频谱

从图 1-22 可见，加汉宁窗、三角窗的信号的频谱的泄漏要比矩形窗的泄漏小很多，通过选择合适的窗函数，达到抑制泄漏的目的。

[例 10] 设 $x(n)=5(0.8)^n \quad 0 \leqslant n \leqslant 20$

（1）分解 $x(n)$成 $x_{ec}(n)$和 $x_{oc}(n)$（偶、奇部分）；

（2）检验实序列的性质：

$\mathrm{DFT}[x_{ec}(n)]=\mathrm{Re}[X(k)]$

$\mathrm{DFT}[x_{oc}(n)]=\mathrm{Im}\,[X(k)]$

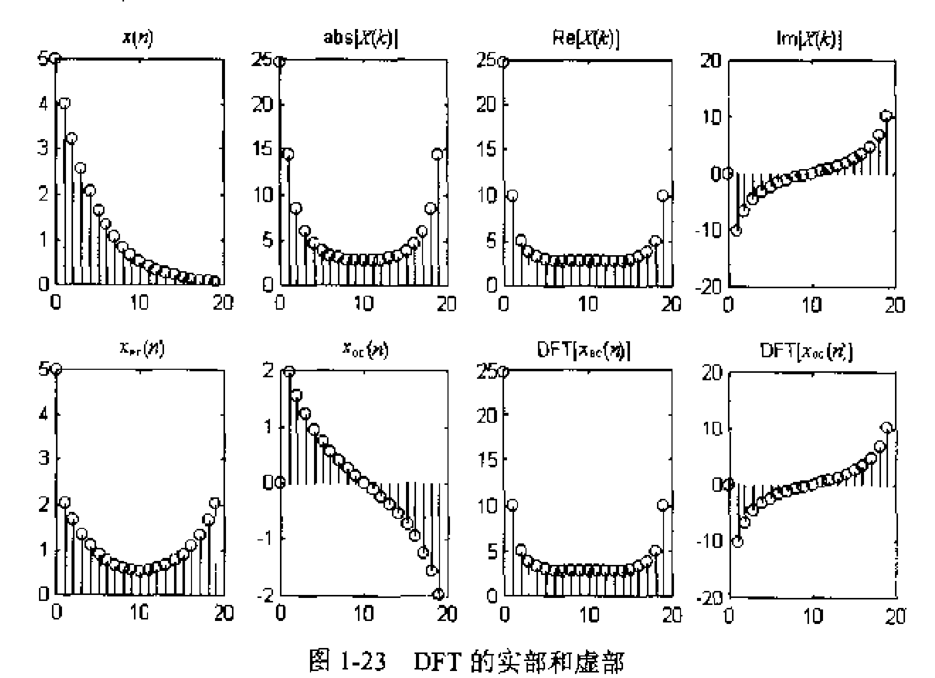


图 1-23 DFT 的实部和虚部

用 MATLAB 实现程序如下，结果见图 1-23。

```
%s121.m
N=20;n=0:N-1;x=5*(0.8).^n;
[xec,xoc]=circevod(x);
X=dft(x,N);Xec=dft(xec,N);Xoc=dft(xoc,N);
subplot(241);stem(n,x);title('x(n)');
subplot(242);stem(n,abs(X));title('abs[X(k)]');
subplot(243);stem(n,real(X));title('Re[X(k)]');
subplot(244);stem(n,imag(X));title('Im[X(k)]');
subplot(245);stem(n,xec);title('xec(n)');
subplot(246);stem(n,xoc);title('xoc(n)');
subplot(247);stem(n,real(Xec));title('DFT[xec(n)]');
subplot(248);stem(n,imag(Xoc));title('DFT[xoc(n)]');
   function [xec,xoc]=circevod(x) %实序列的奇偶分解
   if any(imag(x)~=0)
      error('x 非实数序列')
   end
   N=length(x);n=0:N-1;
   xec=0.5*(x+x(mod(-n,N)+1));
   xoc=0.5*(x-x(mod(-n,N)+1));
```

从此例可见，实序列的偶分量关于 $N/2$ 点对称，奇分量关于 $N/2$ 点反对称；偶分量的 DFT 等于实序列的 DFT 的实部，奇分量的 DFT 等于实序列的 DFT 的虚部。

1.3.4 快速傅里叶变换

DFT 是信号分析与处理中的一种重要变换。但直接计算 DFT 的运算量与变换的长度 N 的平方成正比，当 N 较大时，计算量太大。在快速傅里叶变换（简称 FFT）出现以前，直接用 DFT 算法进行谱分析和信号的实时处理是不实际的。FFT 使得 DFT 的运算效率大大提高，为数字信号处理技术应用于各种信号的实时处理创造了条件，推动了数字信号处理技术的发展。

（1）基 2FFT 算法

长度为 N 的有限长序列 $x(n)$的 DFT 为

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn} \quad k=0,1,\cdots,N-1 \tag{1-13}$$

利用 $W_N^{kn}=W_N^{k(n+N)}=W_N^{(k+N)n}$ 和 $W_N^{kn+\frac{N}{2}}=-W_N^{kn}$，将 N 点 DFT 转化为 $N/2$ 点的 DFT。

当 $N=2^\gamma$ 时，将 $x(n)$按奇偶分成 $N/2$ 的序列：

$$x_1(r)=x(2r) \qquad 0\leqslant r\leqslant \frac{N}{2}-1$$

$$x_2(r)=x(2r+1) \qquad 0\leqslant r\leqslant \frac{N}{2}-1$$

则

$$\begin{aligned}X(k)&=\sum_{n=0}^{N-1}x(n)W_N^{kn}=\sum_{r=0}^{\frac{N}{2}-1}x(2r)W_N^{k2r}+\sum_{n=0}^{\frac{N}{2}-1}x(2r+1)W_N^{k(2r+1)}\\&=X_1(k)+W_N^kX_2(k) \qquad k=0,1,\cdots,N-1\end{aligned}$$

式中

$$X_1(k)=\sum_{r=0}^{N/2-1}x_1(r)W_{N/2}^{kr}=\mathrm{DFT}[x_1(r)]$$

$$X_2(k)=\sum_{r=0}^{N/2-1}x_2(r)W_{N/2}^{kr}=\mathrm{DFT}[x_2(r)]$$

由于 $X_1(k)$和 $X_2(k)$都是 $N/2$ 点的 DFT，且 $W_N^{kn+\frac{N}{2}}=-W_N^{kn}$，所以 $X(k)$也可表示为

$$X(k)=X_1(k)+W_N^kX_2(k) \qquad k=0,1,\cdots,\frac{N}{2}-1$$

$$X\left(k+\frac{N}{2}\right)=X_1(k)-W_N^kX_2(k) \qquad k=0,1,\cdots,\frac{N}{2}-1$$

这样就将 N 点 DFT 分解为两个 $N/2$ 点的 DFT 的运算。$N/2$ 点的 DFT 还可以再分解为 $N/4$ 点的 DFT，共可以分解γ级，最后达到 $N/2$ 个 2 点 DFT 运算。这种方法由于每一步都是按每级输入序列在时间上的次序是属于偶数还是奇数来分解为两个更短的子序列，所以称为“按时间抽取法”。例如 $N=8$ 时，其信号流图见图 1-24。

（2）FFT 算法的运算量

直接计算 DFT 共需

复乘法数 $m=N^2$

复加法数 $a=N(N-1)$

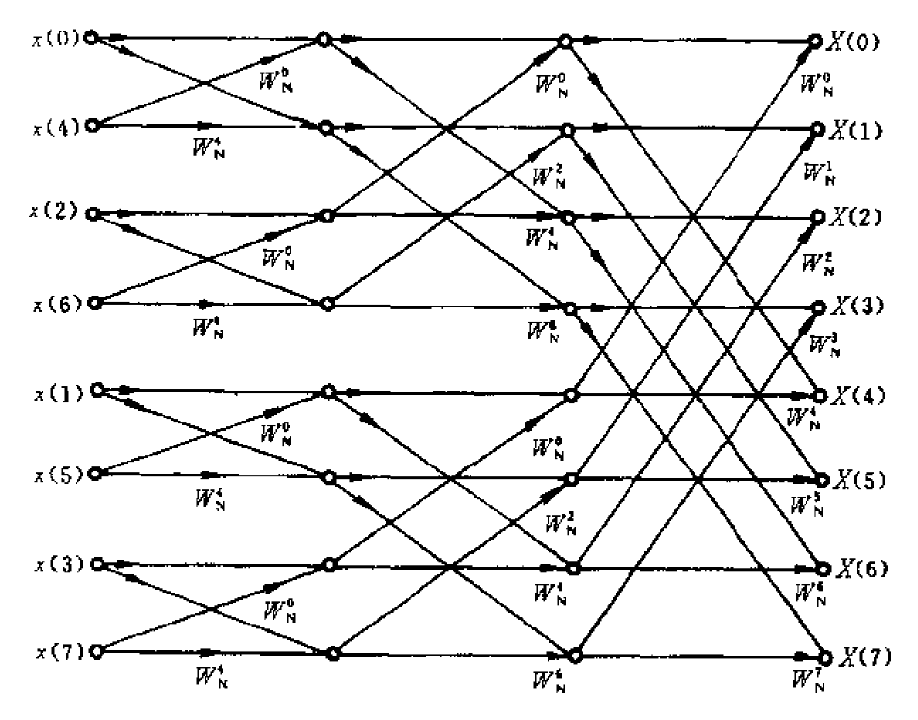


图 1-24 N=8 时的按时间抽取的流图

计算 FFT 共需

$$\text{复乘法数}\ a=\frac{N}{2}\gamma=\frac{N}{2}\log_2 N$$

$$\text{复加法数}\ a=N\gamma=N\log_2 N$$

由于计算机上乘法所需时间比加法所需时间多得多，故直接计算 DFT 与计算 FFT 的运算量之比为

$$\frac{N^2}{\frac{N}{2}\gamma}=\frac{2N}{\log_2 N}$$

具体数据见表 1-3。运算量比较如图 1-25 所示。

表 1-3 DFT 与 FFT 算法的比较

N	N^2	$\frac{N}{2}\log_2 N$	$N^2\Big/\left(\frac{N}{2}\log_2 N\right)$
2	4	1	4.0
4	16	4	4.0
8	64	12	5.4
16	256	32	8.0
32	1024	80	12.8
64	4096	192	21.4
128	16384	448	36.6
256	65536	1024	64.0
512	262144	2304	113.8
1024	1048576	5120	204.8
2048	4194304	11264	372.4

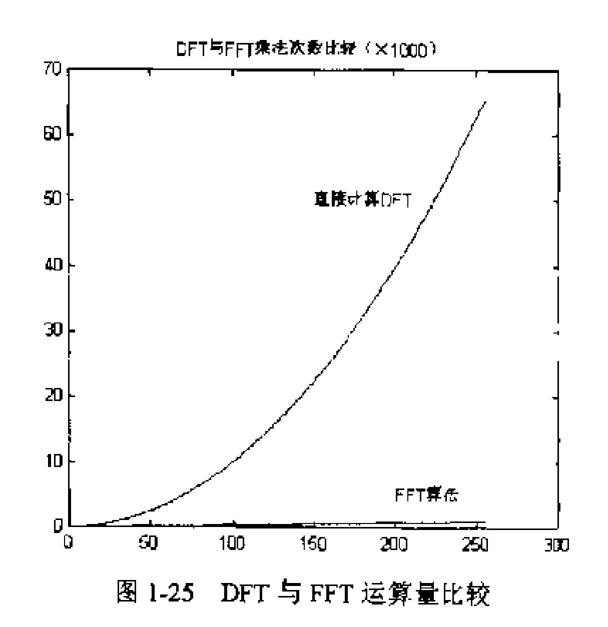


图 1-25 DFT 与 FFT 运算量比较

（3）离散傅里叶反变换的快速算法（IFFT）

由

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn} \qquad n=0,1,\cdots,N-1 \tag{1-14}$$

只要把 DFT 中的每一个系数 W_N^{kn} 换成 W_N^{-kn}，把 $x(n)$换成 $X(k)$，而 $X(k)$换成 $x(n)$每列乘 $\frac{1}{2}$ 因子 $\left[\text{由于}\frac{1}{N}=\left(\frac{1}{2}\right)^{\gamma}\text{，相当每列乘}\frac{1}{2}\text{因子}\right]$，则可用基 2 的 FFT 计算 IFFT。

对式（1-14）取共轭

$$x^*(n)=\frac{1}{N}\sum_{k=0}^{N-1}X^*(k)W_N^{kn} \tag{1-15}$$

对式（1-15）再取共轭得

$$x(n)=\frac{1}{N}\left[\sum_{k=0}^{N-1}X^*(k)W_N^{kn}\right]^*$$

只要先对 $X(k)$取共轭，就可以利用 FFT 计算，再将结果取一次共轭，并乘以 $\frac{1}{N}$。

（4）FFT 算法的 MATLAB 实现

MATLAB 提供 fft 函数来计算 $x(n)$ 的 DFT，fft 函数是用机器语言，而不是以 MATLAB 指令写成的，因此它的执行速度很快。

格式：y=fft(x)，计算信号 x 的快速离散傅里叶变换 y。当 x 为矩阵（多通道信号）时，计算 x 中每一列信号的离散傅里叶变换。当 x 的长度为 2 的幂时，用基 2 算法，否则采用较慢的分裂基算法。

y=fft(x,n)，计算 n 点 FFT，当 x 的长度大于 n 时，截断 x，否则补零。

IFFT 可由 ifft 函数来计算。

1.3.5　快速傅里叶变换应用实例

（1）用 FFT 分析信号频率成分

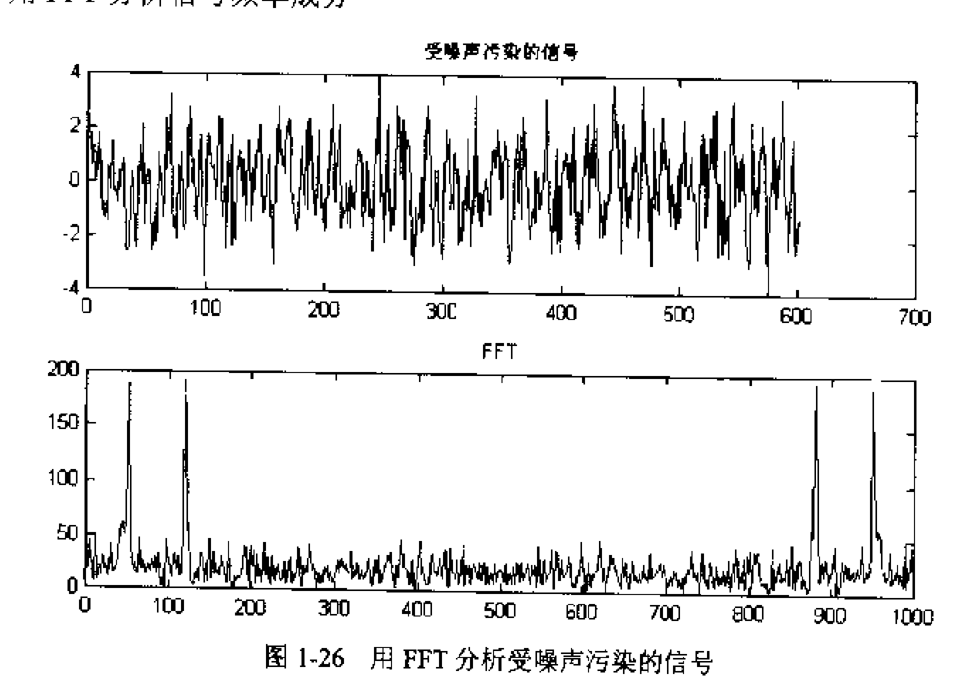


图 1-26　用 FFT 分析受噪声污染的信号

一被噪声污染的信号，很难看出它所包含的频率分量，如一个由 50Hz 和 120Hz 正弦信号构成的信号，受到均值随机噪声的干扰，数据采样率为 1000Hz。通过 FFT 来分析其信号频率成分，用 MATLAB 实现如下。

```
%s123.m
t=0:0.001:0.6;
x=sin(2*pi*50*t)+sin(2*pi*120*t);
y=x+randn(1,length(t));
Y=fft(y,512);
subplot(211);plot(x);title('受噪声污染的信号');
n=0:511;f=1000*n/512;
subplot(212);plot(f,abs(Y));title('FFT');
```

从图 1-26 可见信号集中在 50Hz 和 120Hz。

（2）用 FFT 分析语音信号的频谱

结果见图 1-27，用 MATLAB 实现如下。

```
%s124.m
load mtlb;
subplot(221);plot(mtlb);title('原始语音信号');
y=fft(mtlb);
subplot(222);plot(abs(y));title('FFT 变换');
y(abs(y)<1)=0;x=ifft(y);
subplot(223);plot(abs(y));title('去掉幅值小于 1 的 FFT 变换值');
subplot(224);plot(real(x));title('重构语音信号');
```

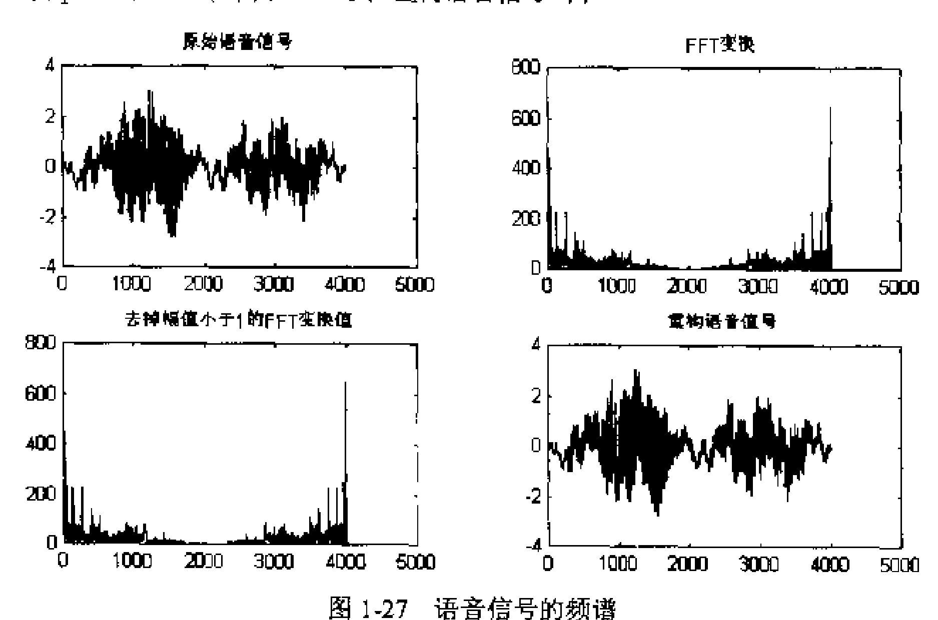


图 1-27　语音信号的频谱

1.3.6　离散余弦变换（DCT）

离散傅里叶变换是频域分析的有效工具，但它的变换核是复指数，运算量太大，难以实时处理。在寻求 DFT 快速算法的同时，人们构造出了一种实数域变换——离散余弦变换，简称 DCT。DCT 的变换核为实数的余弦函数，计算速度较快。

给定有限长离散信号 $x(n)$，其 DCT 定义为

$$X_c(0)=\frac{1}{N}\sum_{n=0}^{N-1}x(n)$$
$$X_c(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}x(n)\cos\frac{(2n+1)k\pi}{2N}\qquad k=1,2,\cdots N-1$$

离散余弦反变换（IDCT）定义为

$$x(n)=\frac{1}{N}X_c(0)+\sqrt{\frac{2}{N}}\sum_{k=0}^{N-1}X_c(k)\cos\frac{(2n+1)k\pi}{2N}\qquad n=0,1,\cdots N-1$$

[例 11]　计算信号 $x(n)=n+50\cos\left(\frac{2\pi n}{40}\right),\qquad n=1,2,\cdots,100$ 的 DCT。

```
%s125.m
N=100;n=1:N;
x=n+50*cos(2*pi*n/40);
y=dct(x);
subplot(121);sten(x);title('原始信号');
subplot(122);stem(y);title('DCT 变换');
```

从图 1-28 可见，在 DCT 变换域上，信号的能量主要集中在几个变化系数上，这一特点是用 DCT 变换进行数据压缩的基本依据。

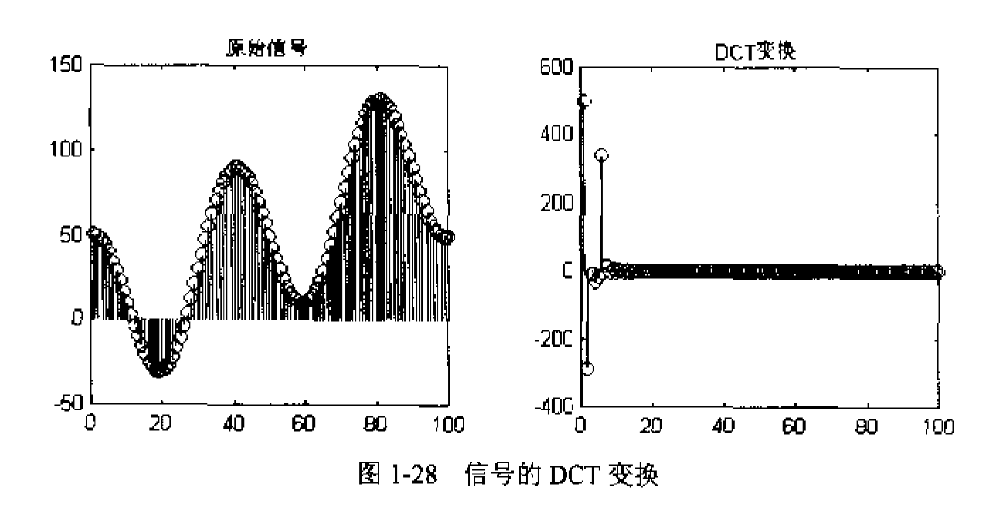


图 1-28　信号的 DCT 变换

[例 12]　用 DCT 变换进行语音压缩，结果见图 1-29。

```
%s126.m
load mtlb;
subplot(221);plot(mtlb);title('原始语音信号');
y=dct(mtlb);
subplot(222);plot(y);title('DCT 变换');
y(abs(y)<1)=0;x=idct(y);
subplot(223);plot(y);title('去掉幅值小于 1 的 DCT 变换值');
subplot(224);plot(x);title('重构语音信号');
```

与离散傅里叶变换相比，信号的离散余弦变换避免了繁杂的复数运算，实信号的 DCT 变换还是实数。它具有更好的能量压缩性能，仅用少数几个变换系数就可表征信号的总体。这种性质使得它在数据压缩、图像压缩和数据通信中得到广泛的应用。

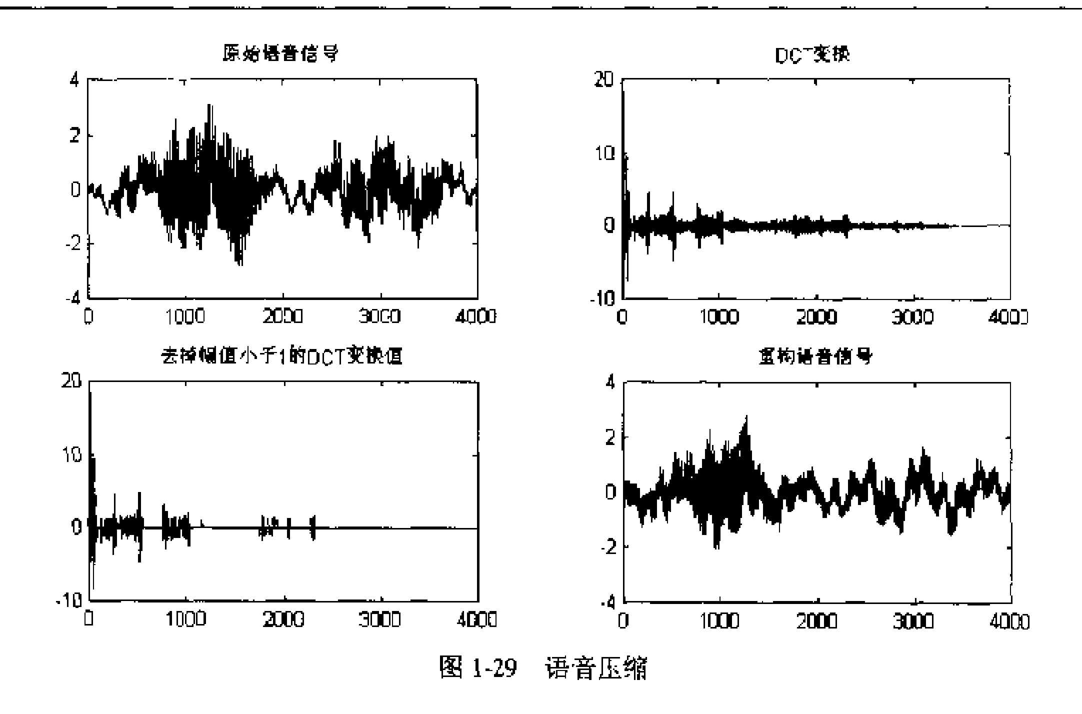


图 1-29 语音压缩

1.3.7 线性调频 z 变换（CZT）

离散傅里叶变换可以看作信号在 z 域上沿单位圆的均匀采样，但在实际应用中，并非整个单位圆上的频谱都有意义。例如，对于窄带信号，只分析信号所在的一段频带。另外，有时采样点也不限于单位圆。例如，语音信号处理中经常需要知道极点所对应的频率，若极点位置离单位圆较远，则其频谱就会很光滑，难以构成峰值，这样就需要采样点的轨迹是一条接近这些极点的弧线或圆周。

已知 $x(n)$, $0\leqslant n\leqslant N-1$，它的 z 变换为

$$X(z)=\sum_{n=0}^{N-1}x(n)z^{-n}$$

令 z 的取样点 z_k 为

$$z_k=AW^{-k}\qquad k=0,1,\cdots,M-1$$

A 和 W 是任意复数，即

$$A=A_0\mathrm{e}^{\mathrm{j}\theta_0}$$

$$W=W_0\mathrm{e}^{-\mathrm{j}\varphi_0}$$

式中 M—— 欲分析的复频谱的点数，不一定等于 N；

A—— 取样轮廓线的复数起点；

W—— 取样轮廓线上各点之间的比率；

A_0—— 取样起始样点的矢量半径；

W_0—— 控制轮廓线向内还是向外弯曲；

θ_0—— 起始样点的相角；

φ_0—— 相邻样点的角频率差。

定义信号 $x(n)$的 CZT 为

$$X(z_k)=\sum_{n=0}^{N-1}x(n)A^{-n}W^{kn}\qquad 0\leqslant k\leqslant M-1$$

利用布鲁斯坦等式

$$nk = \frac{1}{2}[n^2 + k^2 - (k-n)^2]$$

则

$$X(z_k) = W^{\frac{k^2}{2}} \sum_{n=0}^{N-1}\left[x(n)A^{-n}W^{\frac{n^2}{2}}\right]W^{\frac{-(k-n)^2}{2}} \qquad 0 \leqslant k \leqslant M-1$$

[例 13] 已知一信号

$$x(n) = \begin{cases} 1 & 0 \leqslant n \leqslant 9 \\ 0 & 其他 \end{cases}$$

用 CZT 求其前 30 点的复频谱 $X(z_k)$，给定 A_0=0.8，W_0=1.2，$\theta_0 = \frac{\pi}{3}$，$\varphi_0 = \frac{2\pi}{20}$。

```
%s128.m
A=0.8*exp(j*pi/3);
W=1.2*exp(-j*pi*0.1);
M=30;
zk=A*(W.^(-(0:M-1)));
subplot(131);zplane(zk.');
x=[1,1,1,1,1,1,1,1,1,1];
z=czt(x,M,W,A);k=0:M-1;n=0:9;
subplot(132);stem(k,abs(z));title('CZT');
y=fft(x,M);
subplot(133);stem(k,abs(y));title('FFT');
```

结果如图 1-30 所示。

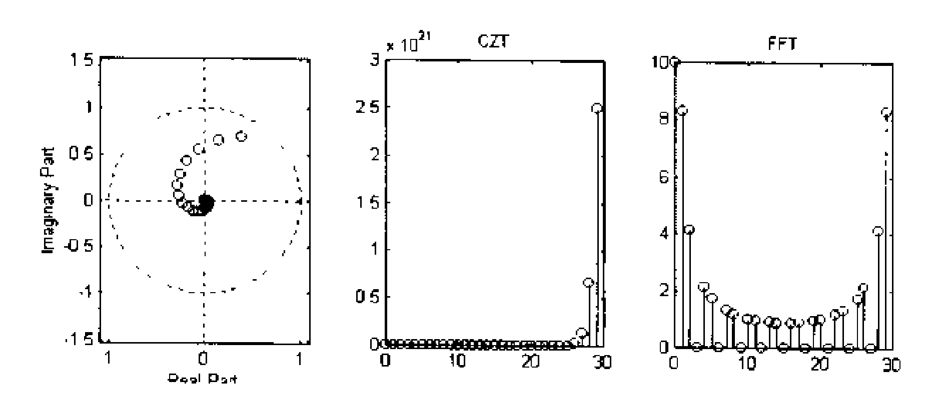


图 1-30 线性调频 z 变换

利用 CZT 计算滤波器 x 在 100～150Hz 频率特性的细节。

```
%s127.m
x=fir1(30,125/500,boxcar(31));
Fs=1000;
f1=100;f2=150;
m=1024;
w=exp(-j*2*pi*(f2-1)/(m*Fs));
a=exp(j*2*pi*f1/Fs);
y=fft(x,1000);
```

```
z=czt(x,m,w,a);
fy=0:(length(y)-1)'*1000/length(y);
fz=((0:length(z)-1)'*(f2-f1)/length(z))+f1;
subplot(121);plot(fy(1:500),abs(y(1:500)));title('FFT');
subplot(122);plot(fz,abs(z));title('CZT');
```

结果如图 1-31 所示。

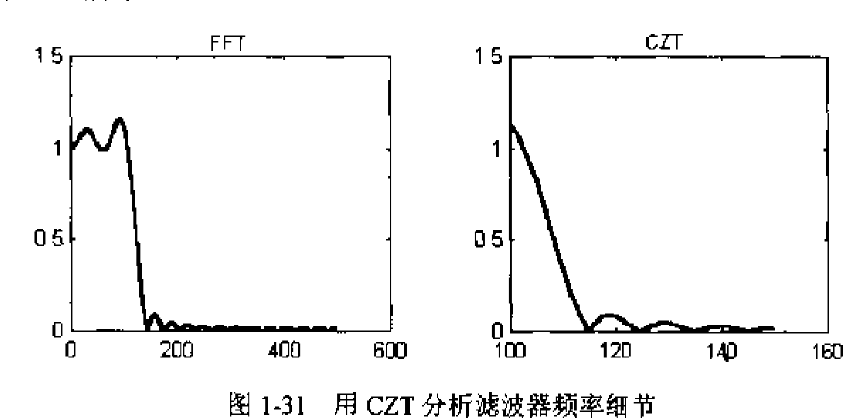


图 1-31　用 CZT 分析滤波器频率细节

第 2 章　数字滤波器设计及实现

2.1　数字滤波器结构

为了处理信号，必须设计和实现数字滤波器系统。滤波器的设计结果受滤波器的类型（IIR 或 FIR）和其实现形式（结构）的影响。所以在讨论设计结果以前，先考虑在实际中如何实现这些滤波器。

滤波器的结构按单位脉冲响应的性质可分为：无限长单位脉冲响应滤波器（IIR）和有限长单位脉冲响应滤波器（FIR）。

2.1.1　IIR 滤波器结构

一个 IIR 滤波器的系统函数可表示为

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{i=0}^{M}b_i z^{-i}}{1-\sum_{k=1}^{N}a_k z^{-k}}$$

对应的差分方程为

$$y(n)=\sum_{i=0}^{M}b_i x(n-i)+\sum_{k=1}^{N}a_k y(n-k)$$

其中 a_k、b_i 是滤波器的系数。

无限长单位脉冲响应滤波器有以下特点：

① 系统的单位脉冲响应 $h(n)$是无限长的。

② 系统函数 $H(z)$在有限 z 平面上有极点存在。

③ 结构上有反馈。

为了用计算机或专用硬件实现数字滤波器，必须把上两式变换成一种算法，按照这种算法对输入信号进行运算，得到输出信号。而差分方程就是一种具体算法：系统输出等于系统输入的各延迟信号与输出各延迟信号的线性组合。

描述滤波器结构的基本元件有以下几种。

① 加法器：此元件有两个输入和一个输出。

② 乘法器：此元件是单输入单输出的。

③ 延迟元件：此元件把通过它的信号延迟一个样本。

IIR 滤波器结构分成直接型、级联型和并联型。

（1）直接型

直接型系统函数可表示为

$$H(z)=\frac{b_0+b_1 z^{-1}+\cdots+b_M z^{-M}}{1+a_1 z^{-1}+\cdots+a_N z^{-N}}$$

在 MATLAB 中，用 filter 实现 IIR 的直接形式。

［例 1］ 用直接型实现系统函数为 $H(z)=\dfrac{1-3z^{-1}+11z^{-2}+27z^{-3}+18z^{-4}}{1+16z^{-1}+12z^{-2}+2z^{-3}-4z^{-4}-z^{-5}}$ 的 IIR 数字滤波器，求单位脉冲响应和单位阶跃信号的输出。

```
%s131.m
b=[1,-3,11,27,18];a=[16,12,2,-4,-1];
N=25;delta=impseq(0,0,N);
h=filter(b,a,delta);%直接型单位脉冲响应
x=[ones(1,5),zeros(1,N-5)];%单位阶跃信号
y=filter(b,a,x);%直接型输出信号
subplot(121);stem(h);title('直接型 h(n)');
subplot(122);stem(y);title('直接型 y(n)');
```

结果见图 2-1。

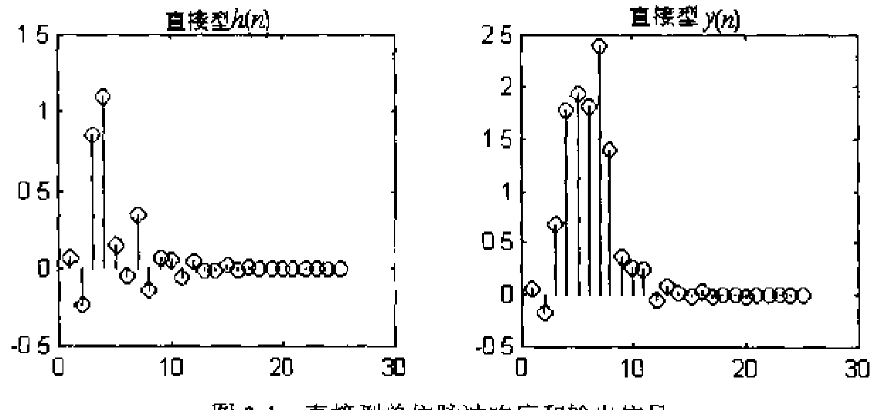


图 2-1 直接型单位脉冲响应和输出信号

（2）级联型

级联型系统函数可表示为

$$H(z)=b_0\prod_{k=1}^{K}\frac{1+B_{k,1}z^{-1}+B_{k,2}z^{-2}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}}$$

在 MATLAB 中给定级联型系统函数，由扩展函数 casfiltr 实现 IIR 的级联形式。

［例 2］ 系统函数 $H(z)=\dfrac{4(1+z^{-1})(1-1.4142136z^{-1}+z^{-2})}{(1-0.5z^{-1})(1+0.9z^{-1}+0.81z^{-2})}$，用级联型结构实现。

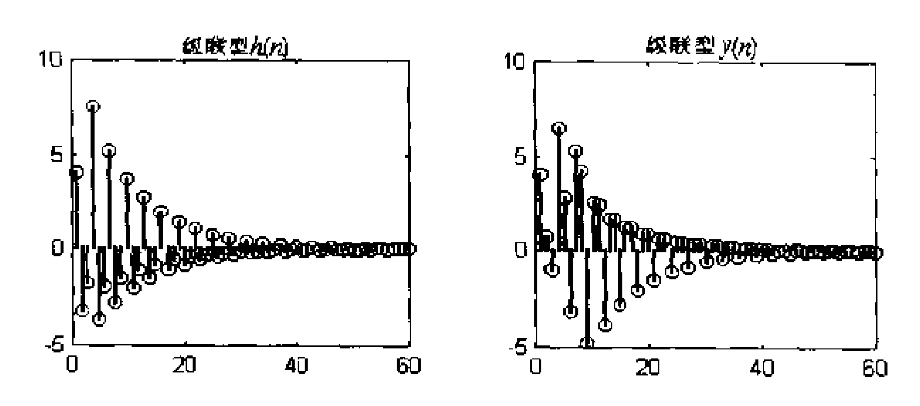


图 2-2 级联型单位脉冲响应和输出信号（例 2）

```
%s132.m
```

```
b0=4;B=[1,1,0;1,-1.4142136,1];A=[1,-0.5,0;1,0.9,0.81];
N=60;delta=impseq(0,0,N);
h=casfiltr(b0,B,A,delta);
x=[ones(1,5),zeros(1,N-5)];
y=casfiltr(b0,B,A,x);
subplot(121);stem(h);title('级联型h(n)');
subplot(122);stem(y);title('级联型y(n)');
```

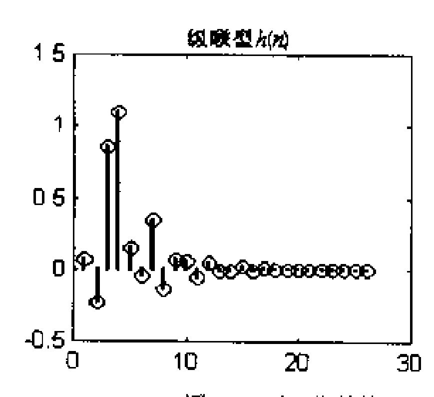
结果见图 2-2。

● 若给定直接形式滤波器的系数 a_k、b_i，可由扩展函数 dir2cas 计算出 b_0、$B_{k,i}$ 和 $A_{k,i}$，然后由扩展函数 casfiltr 实现 IIR 的级联形式。

［例 3］ 用级联型实现系统函数为 $H(z)=\dfrac{1-3z^{-1}+11z^{-2}+27z^{-3}+18z^{-4}}{1+16z^{-1}+12z^{-2}+2z^{-3}-4z^{-4}-z^{-5}}$ 的 IIR 数字滤波器，求单位脉冲响应和单位阶跃信号的输出。

```
%s133.m
N=25;delta=impseq(0,0,N);
b=[1,-3,11,27,18];a=[16,12,2,-4,-1];
[b0,B,A]=dir2cas(b,a);%直接型转换成级联型
h=casfiltr(b0,B,A,delta);%级联型单位脉冲
x=[ones(1,5),zeros(1,N-5)];
y=casfiltr(b0,B,A,x);%级联型输出信号
subplot(121);stem(h);title('级联型h(n)');
subplot(122);stem(y);title('级联型y(n)');
```

结果见图 2-3。



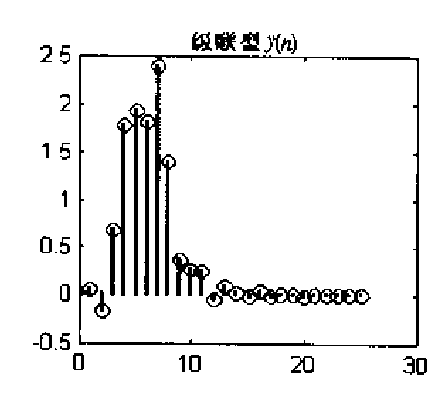


图 2-3 级联型单位脉冲响应和输出信号（例 3）

系统函数为 $H(z)=\dfrac{0.0625(1+1.7366z^{-1}+0.9869z^{-2})(1+1.0000z^{-1}+0.5000z^{-2})}{(1-4.7366z^{-1}+18.2390z^{-2})(1-0.2500z^{-1}-0.1250z^{-2})}$，从例 1 和例 3 的结果可见：直接型和级联型的单位脉冲响应、输出信号是一样的。

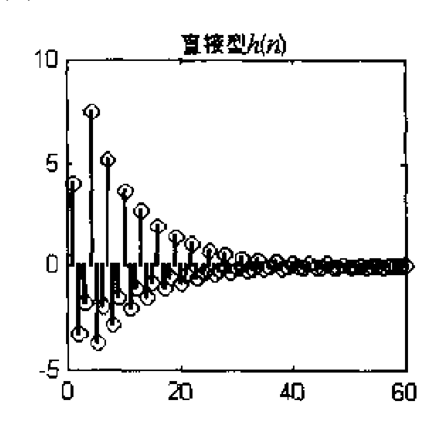
若给定系统函数为级联形式时，也可由扩展函数 cas2dir 将其转换成直接形式，再由 filter 直接形式实现。

［例 4］ 用直接型结构实现系统函数为 $H(z)=\dfrac{4(1+z^{-1})(1-1.4142136z^{-1}+z^{-2})}{(1-0.5z^{-1})(1+0.9z^{-1}+0.81z^{-2})}$ 的 IIR 数字滤波器，求单位脉冲响应和单位阶跃信号的输出。

```
%s134.m
```

```
b0=4;B=[1,1,0;1,-1.4142136,1];A=[1,-0.5,0;1,0.9,0.81];
N=60;delta=impseq(0,0,N);
x=[ones(1,5),zeros(1,N-5)];
[b,a]=cas2dir(b0,B,A);
h=filter(b,a,delta);
y=filter(b,a,x);
subplot(121);stem(h);title('直接型 h(n)');
subplot(122);stem(y);title('直接型 y(n)');
```

结果见图 2-4。



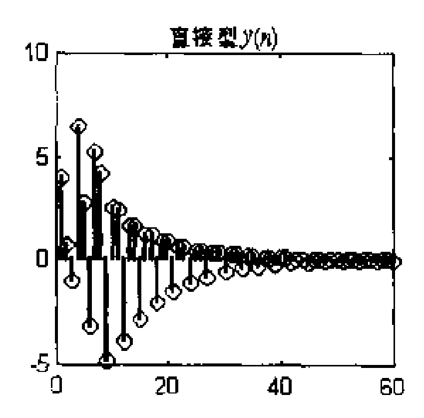


图 2-4 直接型单位脉冲响应和输出信号（例 4）

直接型系统函数为 $H(z)=\dfrac{4-1.6569z^{-1}-1.6569z^{-2}+4z^{-3}}{1+0.4000z^{-1}+0.3600z^{-2}-0.4050z^{-3}}$

（3）并联型

并联型系统函数可表示为

$$H(z)=\sum_{k=1}^{K}\frac{B_{k,0}+B_{k,1}z^{-1}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}}+\sum_{0}^{M-N}C_k z^{-k}$$

在 MATLAB 中用给定并联系统函数，由扩展函数 parfiltr 实现 IIR 的并联形式。

［例 5］ 用并联型实现系统函数为 $H(z)=\dfrac{-14.75-12.90z^{-1}}{1-\frac{7}{8}z^{-1}+\frac{3}{32}z^{-2}}+\dfrac{24.50+26.82z^{-1}}{1-z^{-1}+\frac{1}{2}z^{-2}}$ 的滤波器。

```
%s135.m
C=0;B=[-14.75,-12.90;24.50,26.82];A=[1,-7/8,3/32;1,-1,0.5];
N=60;delta=impseq(0,0,N);
h=parfiltr(C,B,A,delta);
x=[ones(1,5),zeros(1,N-5)];
y=parfiltr(C,B,A,x);
subplot(121);stem(h);title('并联型 h(n)');
subplot(122);stem(y);title('并联型 y(n)');
```

结果见图 2-5。

若给定直接形式滤波器的系数 a_k、b_i，可由扩展函数 dir2par 计算出 b_0、$B_{k,i}$ 和 $A_{k,i}$，然后由扩展函数 parfiltr 实现 IIR 的并联形式。

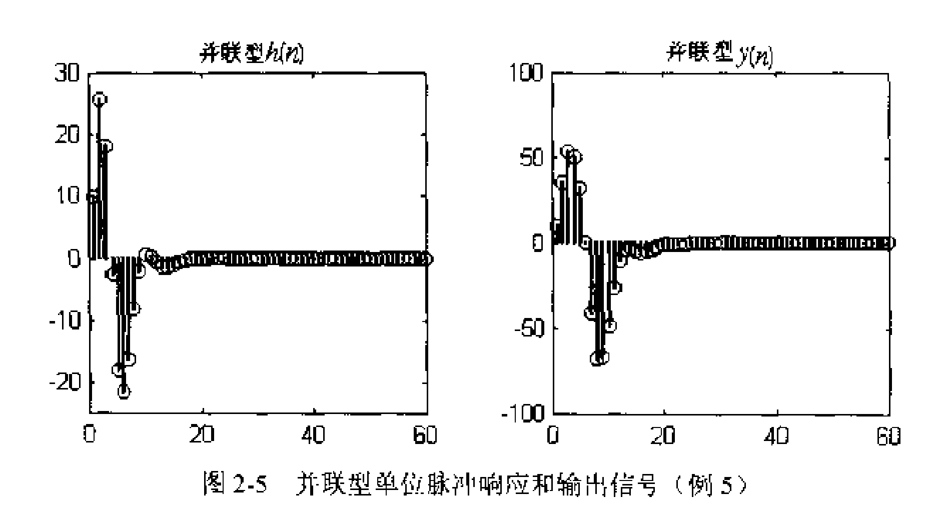


图 2-5　并联型单位脉冲响应和输出信号（例 5）

[例 6]　用并联型实现系统函数为 $H(z)=\dfrac{1-3z^{-1}+11z^{-2}+27z^{-3}+18z^{-4}}{1+16z^{-1}+12z^{-2}+2z^{-3}-4z^{-4}-z^{-5}}$ 的 IIR 数字滤波器，求单位脉冲响应和单位阶跃信号的输出。

```
%s136.m
b=[1,-3,11,27,18];a=[16,12,2,-4,-1];
N=25;delta=impseq(0,0,N);
[C,B,A]=dir2par(b,a);
h=parfiltr(C,B,A,delta);
x=[ones(1,5),zeros(1,N-5)];
y=parfiltr(C,B,A,x);
subplot(121);stem(h);title('并联型 h(n)');
subplot(122);stem(y);title('并联型 y(n)');
```

结果见图 2-6。

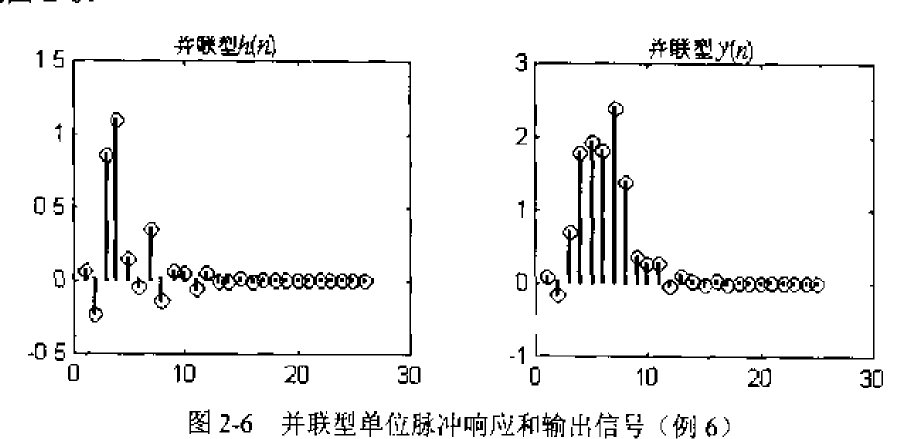


图 2-6　并联型单位脉冲响应和输出信号（例 6）

并联型系统函数为 $H(z)=\dfrac{0.75+1.45z^{-1}}{1+z^{-1}+0.5z^{-2}}+\dfrac{17.3125-5.2625z^{-1}}{1-0.2500z^{-1}-0.1250z^{-2}}-18$

若给定系统函数为并联形式时，也可由扩展函数 par2dir 将其转换成直接形式，再由 filter 直接形式实现。

[例 7]　用直接型实现系统函数 $H(z)=\dfrac{-14.75-12.90z^{-1}}{1-\dfrac{7}{8}z^{-1}+\dfrac{3}{32}z^{-2}}+\dfrac{24.50+26.82z^{-1}}{1-z^{-1}+\dfrac{1}{2}z^{-2}}$ 的滤波器。

```
%s137.m
```

```
C=0;B=[-14.75,-12.90;24.50,26.82];A=[1,-7/8,3/32;1,-1,0.5];
N=60;delta=impseq(0,0,N);
[b,a]=par2dir(C,B,A);
h=filter(b,a,delta);
x=[ones(1,5),zeros(1,N-5)];
y=filter(b,a,x);
subplot(121);stem(h);title('直接型 h(n)');
subplot(122);stem(y);title('直接型 y(n)');
```

结果见图 2-7。

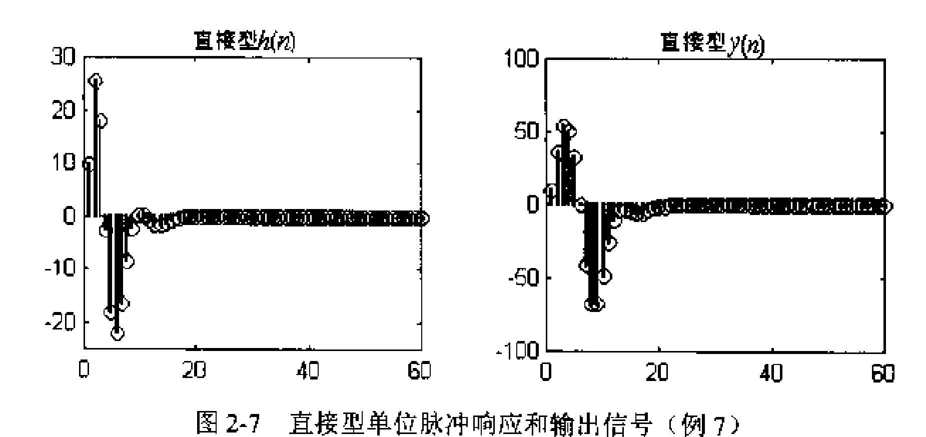


图 2-7　直接型单位脉冲响应和输出信号（例 7）

直接型系统函数为 $H(z)=\dfrac{9.7500+7.2325z^{-1}-15.6456z^{-2}-3.9350z^{-3}}{1-1.8750z^{-1}+1.4688z^{-2}-0.5313z^{-3}+0.0469z^{-4}}$

[例 8]　如果一个滤波器结构中包括直接、级联和并联的组合，它用这三种形式表示的总特性是什么？

$$H(z)=\left[\frac{(2+4z^{-1})}{(1+z^{-1}+0.9z^{-2})}+\frac{(3+z^{-1})}{(1+0.4z^{-1}-0.4z^{-2})}\right]\times\left[\frac{(0.5+0.7z^{-1})}{(1-z^{-1}+0.8z^{-2})}+\frac{(1.5+2.5z^{-1})}{(1+0.5z^{-1}+0.5z^{-2})}+\frac{(0.8+z^{-1})}{(1-0.5z^{-2})}\right]$$

在这种结构中，包含两个并联子系统的级联形式。第一个并联子系统包含两个双二阶环节，而第二个并联子系统包含三个双二阶环节。需要用 par2dir 函数把每个并联子系统转换成直接形式，然后把它们级联起来。最终的直接形式通过分别对应的分子、分母多项式进行卷积得到。最终的级联和并联形式则可由直接形式导出。

```
%s138.m
N=29;delta=impseq(0,0,N);
C0=0;B1=[2,4;3,1];A1=[1,1,0.9;1,0.4,-0.4];
B2=[0.5,0.7;1.5,2.5;0.8,1];A2=[1,-1,0.8;1,0.5,0.5;1,0,-0.5];
[b1,a1]=par2dir(C0,B1,A1);
[b2,a2]=par2dir(C0,B2,A2);
b=conv(b1,b2);%总的直接型分子
a=conv(a1,a2);%总的直接型分母
[b0,Bc,Ac]=dir2cas(b,a);
[C,B,A]=dir2par(b,a);
h1=real(filter(b,a,delta));
```

```
h2=casfiltr(b0,Bc,Ac,delta);
h3=parfiltr(C,B,A,delta);
subplot(131);stem(h1);title('直接型 h(n)');
subplot(132);stem(h2);title('级联型 h(n)');
subplot(133);stem(h3);title('并联型 h(n)');
```

结果见图 2-8。

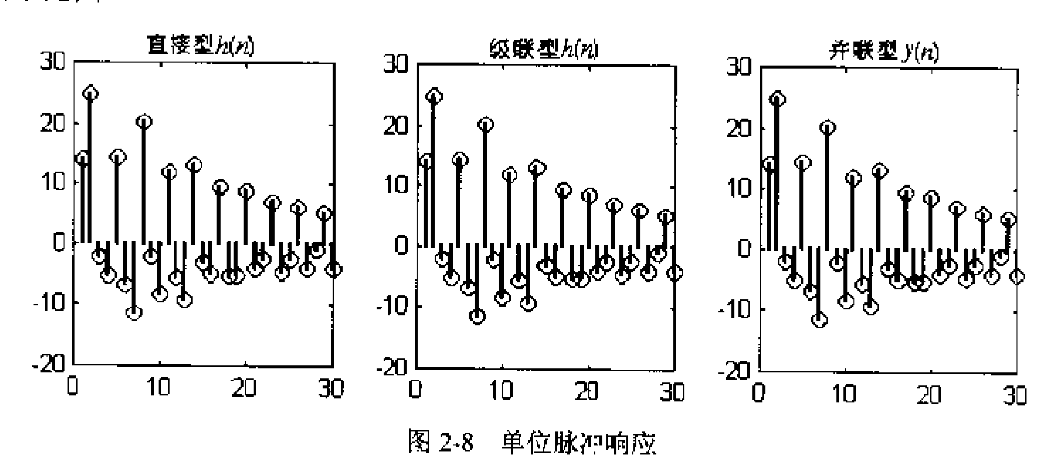


图 2-8 单位脉冲响应

```
%直接型转换为级联型:
function[b0,B,A]=dir2cas(b,a);
b0=b(1);b=b/b0;
a0=a(1);a=a/a0;
b0=b0/a0;
M=length(b);N=length(a);
if N>M
   b=[b zeros(1,N-M)];
elseif M>N
   a=[a zeros(1,M-N)];N=M;
else
   NM=0;
end
K=floor(N/2);B=zeros(K,3);A=zeros(K,3);
if K*2==N;
   b=[b 0];
   a=[a 0];
end
broots=cplxpair(roots(b));
aroots=cplxpair(roots(a));
for i=1:2:2*K
   Brow=broots(i:1:i+1,:);
      Brow=real(poly(Brow));
   B(fix(i+1)/2,:)=Brow;
   Arow=aroots(i:1:i+1,:);
   Arow=real(poly(Arow));
   A(fix(i+1)/2,:)=Arow;
```

```
end
%滤波器级联型实现:
function y=casfiltr(b0,B,A,x);
[K,L]=size(B);
N=length(x);
w=zeros(K+1,N);
w(1,:)=x;
for i=1:K
   w(i+1,:)=filter(B(i,:),A(i,:),w(i,:));
end
y=b0*w(K+1,:);
%级联型转换成直接型:
function[b,a]=cas2dir(b0,B,A);
[K,L]=size(B);
b=[1];
a=[1];
for i=1:K
   b=conv(b,B(i,:));
   a=conv(a,A(i,:));
end
b=b*b0;
%滤波器并联型实现:
function y=parfiltr(C,B,A,x);
[K,L]=size(B);
N=length(x);
w=zeros(K+1,N);
w(1,:)=filter(C,1,x);
for i=1:K
   w(i+1,:)=filter(B(i,:),A(i,:),x);
end
y=sum(w);
%直接型转换成并联型:
function[C,B,A]=dir2par(b,a);
M=length(b);N=length(a);
[r1,p1,C]=residuez(b,a);
p=cplxpair(p1,10000000*eps);
I=cplxcomp(p1,p);
r=r1(I);
K=floor(N/2);B=zeros(K,2);A=zeros(K,3);
if K*2==N;
for i=1:2:N-2
   Brow=r(i:1:i+1,:);
   Arow=p(i:1:i+1,:);
   [Brow,Arow]=residuez(Brow,Arow,[ ]);
   B(fix((i+1)/2),:)=real(Brow);
   A(fix((i+1)/2),:)=real(Arow);
```

```
end
    Brow,Arow]=residuez(r(N-1),p(N-1),[ ]);
   B(K,:)=[real(Brow)0];A(K,:)=[real(Arow)0];
else
   for i=1:2:N-1
   Brow=r(i:1:i+1,:);
   Arow=p(i:1:i+1,:);
   [Brow,Arow]=residuez(Brow,Arow,[ ]);
   B(fix((i+1)/2),:)=real(Brow);
   A(fix((i+1)/2),:)=real(Arow);
end
end
%并联型转换成直接型:
function[b,a]=par2dir(C,B,A);
   [K,L]=size(A);R=[];P=[];
for i=1:K
   [r,p,k]=residuez(B(i,:),A(i,:));
   R=[R;r];P=[P;p];
end
[b,a]=residuez(R,P,C);
b=b(:)';a=a(:)';
```

2.1.2 FIR 滤波器结构

一个 FIR 滤波器的系统函数为

$$H(z)=\sum_{n=0}^{M-1}h(n)z^{-n}$$

有限长单位脉冲响应滤波器有以下特点:

① 系统的单位脉冲响应 $h(n)$在有限个 n 值处不为零。

② 系统函数 $H(z)$在$|z|>0$处收敛，极点全部在$z=0$处，所以是因果系统。

③ 结构上没有反馈。

为了与 IIR 数字滤波器共用一个子程序，可将系统函数用如下形式表示:

$$H(z)=\sum_{n=0}^{M-1}b_n z^{-n}$$

其中 $b_n=h(n)$。

（1）直接型

FIR 直接型系统函数表示为

$$H(z)=b_0+b_1z^{-1}+\cdots+b_{M-1}z^{-(M-1)}$$

直接型 FIR 滤波器可通过 filter 函数实现。

（2）级联型

FIR 级联型系统函数表示为

$$H(z)=b_0\prod_{k=1}^{K}1+B_{k,1}z^{-1}+B_{k,2}z^{-2}$$

级联型 FIR 滤波器可通过 casfiltr 函数实现。

[例 9]　FIR 滤波器的系统函数为

$$H(z)=\begin{cases}0.2^n, & 0\leqslant n\leqslant 5\\ 0, & 其他\end{cases}$$

用直接型和级联型分别实现。

```
%s139.m
n=0:5;b=0.2.^n;
N=30;delta=impseq(0,0,N);
h=filter(b,1,delta);%直接型
x=[ones(1,5),zeros(1,N-5)];
y=filter(b,1,x);%直接型
subplot(221);stem(h);title('直接型 h(n)');
subplot(222);stem(y);title('直接型 y(n)');
[b0,B,A]=dir2cas(b,1)
h=casfiltr(b0,B,A,delta);%级联型
y=casfiltr(b0,B,A,x);%级联型
subplot(223);stem(h);title('级联型 h(n)');
subplot(224);stem(y);title('级联型 y(n)');
```

运行后得到图 2-9 所示结果，说明直接型和级联型结果一致。

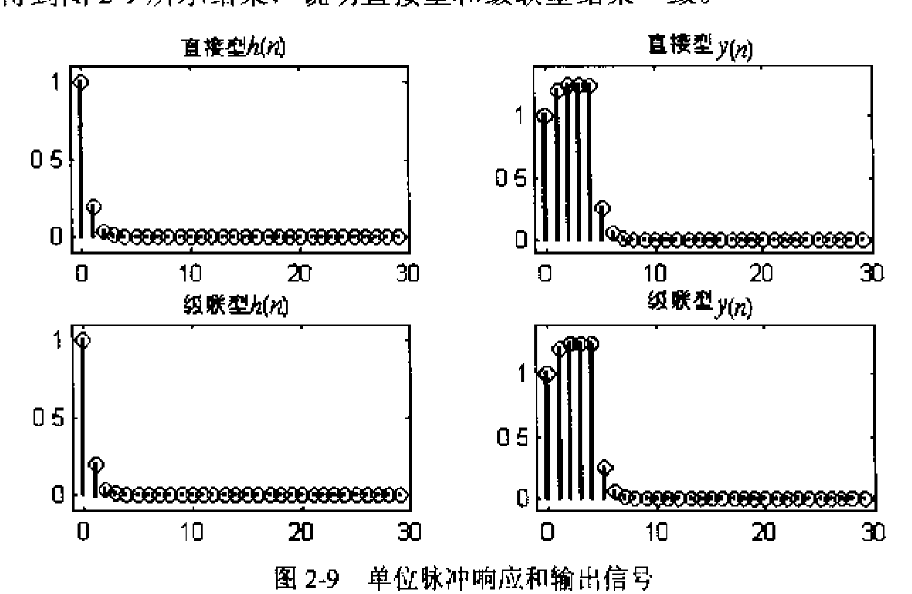


图 2-9　单位脉冲响应和输出信号

```
b0 =1
B =1.0000     0.2000     0.0400
   1.0000    -0.2000     0.0400
   1.0000     0.2000     0
A =1     0     0
   1     0     0
   1     0     0
```

级联形式的系统函数为 $H(z)=(1+0.2z^{-1}+0.04z^{-2})(1-0.2z^{-1}+0.04z^{-2})(1+0.2z^{-1})$

（3）频率采样型

把一个有限长序列的 z 变换 $H(z)$ 在单位圆上做 M 等分采样，即可得到 $H(k)$，则 FIR 系

统函数可以由 $H(k)$ 重构得到

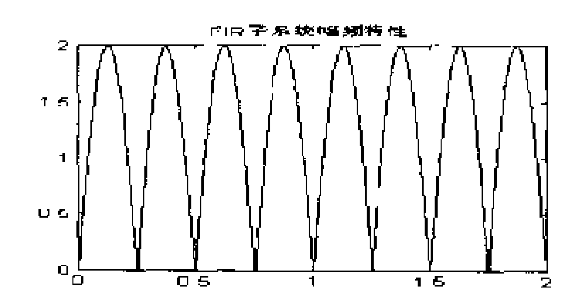
$$H(z)=(1-z^{-M})\frac{1}{M}\sum_{k=0}^{M-1}\frac{H(k)}{1-W_M^{-k}z^{-1}} \tag{2-1}$$

式（2-1）说明在这种结构中，使用的是 $H(k)$，而不是 $h(n)$ 或差分方程。$H(z)$ 包含零点和极点，由它描述的 FIR 滤波器具有与 IIR 滤波器相类似的形式。频率采样型滤波器由一个 FIR 子系统和 IIR 子系统级联构成，它需要采用复数运算。

FIR 子系统 $H_c(z)=1-z^{-M}$

其频率响应为 $H_c(e^{j\omega})=1-e^{-j\omega M}=2\sin\left(\frac{M}{2}\omega\right)e^{-j\omega\frac{M}{2}}$

频率特性见图 2-10。



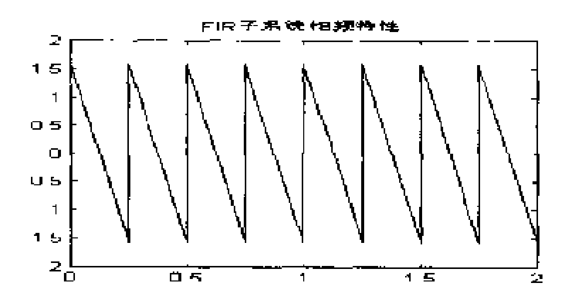


图 2-10　FIR 子系统频率特性

由 FIR 幅频特性曲线(图 2-10)的形状取名为梳状滤波器。其零点为 $z_k=W_M^{k},k=0,1,\cdots,M-1$。

IIR 子系统 $H_k(z)=\dfrac{H(k)}{1-W_M^{k}z^{-1}}$

其极点为 $z_k=W_M^{-k},k=0,1,\cdots,M-1$，正好与 FIR 子系统的零点对消，保持了 FIR 数字滤波器的稳定性。

频率采样型滤波器主要优点：

① 在频率采样点，只要调整一阶网络 $H_k(z)$ 中乘法的系数 $H(k)$，就可有效的调整频响特性，使调整方便。

② 只要 $h(n)$ 的长度相同，对任何频响，其结构完全相同，只是各支路增益不同。相同部分便于标准化、模块化。

由于 FIR 滤波器几乎总是实数，利用 $H(k)$ 的对称性和 W_M^{k} 因子，可把式（2-1）表示成：

$$H(z)=\frac{1-z^{-M}}{M}\left\{\sum_{k=1}^{L}2\,|H(k)|\,H_k(z)+\frac{H(0)}{1-z^{-1}}+\frac{H(M/2)}{1+z^{-1}}\right\} \tag{2-2}$$

其中 M 为奇数时 $L=\dfrac{M-1}{2}$；

M 为偶数时 $L=\dfrac{M}{2}-1$；

$$H_k(z)=\frac{\cos[\angle H(k)]-z^{-1}\cos\left[\angle H(k)-\frac{2\pi k}{M}\right]}{1-2z^{-1}\cos\left(\frac{2\pi k}{M}\right)+z^{-2}} \tag{2-3}$$

若给定直接型的 $h(n)$ 或 $H(k)$，则可以由 dir2fs 函数转换为频率采样结构。

[例 10]　令 $h(n)=\{1,2,3,2,1\}/9$，确定并画出 FIR 的频率取样形式结构。

```
%s140.m
h=[1,2,3,2,1]/9;
[C,B,A]=dir2fs(h)  %直接型的 h(n)转换为频率采样型
```

执行后得

```
C =
    0.5818
    0.0849
    1.0000
B =
   -0.8090    0.8090
    0.3090   -0.3090
A =
    1.0000   -0.6180    1.0000
    1.0000    1.6180    1.0000
    1.0000   -1.0000         0
```

FIR 频率采样结构形式为：

$$H(z)=\frac{1-z^{-5}}{5}\left[0.5818\times\frac{-0.809+0.809z^{-1}}{1-0.618z^{-1}+z^{-2}}+0.0849\times\frac{0.309-0.309z^{-1}}{1+1.618z^{-1}+z^{-2}}+\frac{1}{1-z^{-1}}\right]$$

[例 11]　一个 32 点线性相位 FIR 滤波器的频率样本为

$$|H(k)|=\begin{cases}1, & k=0,1,2\\ 0.5, & k=3\\ 0, & k=4,5,\cdots,15\end{cases}$$

求出它的频率采样形式，计算它的单位脉冲响应 $h(n)$。

解：由于给定 $|H(k)|$ 的样本，利用 $H(k)$ 的对称性和相位约束条件，把 $H(k)$ 重新组合为

$$H(k)=|H(k)|\mathrm{e}^{\mathrm{j}\angle H(k)},k=0,1,\cdots,31$$

$$|H(k)|=|H(32-k)|,k=1,2,\cdots,31;H(0)=1$$

$$\angle H(k)=-\frac{31}{2}\frac{2\pi}{32}k=-\angle H(32-k),k=0,1,\cdots,31$$

则可根据 $H(k)$ 的IDFT求得所希望的单位脉冲响应 $h(n)$。

```
%s141.m
M=32;alpha=(M-1)/2;
magHk=[1,1,1,0.5,zeros(1,25),0.5,1,1];
k1=0:15;k2=16:M-1;
angHk=[-alpha*(2*pi)/M*k1,alpha*(2*pi)/M*(M-k2)];
H=magHk.*exp(j*angHk);
h=real(ifft(H,M));
[C,B,A]=dir2fs(h)
n=0:M-1;subplot(111);stem(n,h);
```

执行后得

```
C =
    2.0000
    2.0000
    1.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    1.0000
         0
B =
   -0.9952    0.9952
    0.9808   -0.9808
   -0.9569    0.9569
   -0.7071    1.0000
   -0.7395   -0.1488
    0.3271    0.7479
   -0.5622   -0.7014
   -0.0000    1.0000
   -0.6136   -0.8941
    0.7058    0.9246
   -0.4668   -0.9947
    0.7071    1.0000
   -0.4472   -0.8688
    1.0000    0.9239
   -0.8000   -0.9017
A =
    1.0000   -1.9616    1.0000
    1.0000   -1.8478    1.0000
    1.0000   -1.6629    1.0000
    1.0000   -1.4142    1.0000
```

1.0000	−1.1111	1.0000
1.0000	−0.7654	1.0000
1.0000	−0.3902	1.0000
1.0000	−0.0000	1.0000
1.0000	0.3902	1.0000
1.0000	0.7654	1.0000
1.0000	1.1111	1.0000
1.0000	1.4142	1.0000
1.0000	1.6629	1.0000
1.0000	1.8478	1.0000
1.0000	1.9616	1.0000
1.0000	−1.0000	0
1.0000	1.0000	0

FIR 频率采样结构形式为

$$H(z)=\frac{1-z^{-32}}{32}\left(2\frac{-0.9952+0.9952z^{-1}}{1-1.9616z^{-1}+z^{-2}}+2\frac{0.9808-0.9808z^{-1}}{1-1.8478z^{-1}+z^{-2}}+\frac{-0.9952+0.9952z^{-1}}{1-1.6629z^{-1}+z^{-2}}+\frac{1}{1-z^{-1}}\right)$$

单位脉冲响应$h(n)$如图2-11所示。

当采样点数 M 很大时，其结构很复杂，需要的乘法和延时单元很多。但对于窄带滤波器，大部分频率采样值为零，从而使二阶网络个数大大减少，所以频率采样结构适用于窄带滤波器。

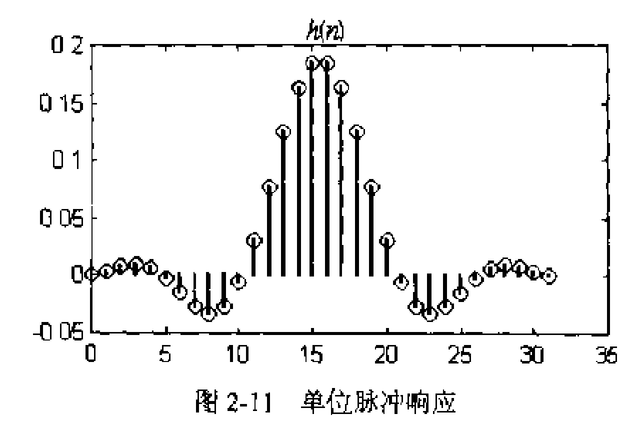


图 2-11 单位脉冲响应

```
%直接型到频率采样型的转换
function[C,B,A]=dir2fs(h)
%C=并联部分增益的行向量
%B=按行排列的分子系数矩阵
%A=按行排列的分母系数矩阵
%h= FIR 滤波器的单位脉冲响应向量
M=length(h);
H=fft(h,M);
magH=abs(H);phaH=angle(H)';
if(M==2*floor(M/2))
   L=M/2-1;
   A1=[1,-1,0;1,1,0];
   C1=[real(H(1)),real(H(L+2))];
else
   L=(M-1)/2;
   A1=[1,-1,0];
   C1=[real(H(1))];
end
k=[1:L]';
B=zeros(L,2);A=ones(L,3);
A(1:L,2)=-2*cos(2*pi*k/M);A=[A;A1];
```

```
B(1:L,1)=cos(phaH(2:L+1));
B(1:L,2)=-cos(phaH(2:L+1)-(2*pi*k/M));
C=[2*magH(2:L+1),C1]';
```

2.2 FIR 滤波器设计

在数字信号处理中有两种类型的重要系统。第一种系统在时域中进行信号滤波，因此叫数字滤波器；第二种系统提供频域里的信号表示，叫频谱分析仪。

所谓数字滤波是指通过一种数值运算，达到改变输入信号中所含频率分量的相对比例，或者滤除某些频率分量。因此，数字滤波器和模拟滤波器有同样的滤波概念，区别是实现方式不同。由于数字滤波采用数值运算的方法达到滤波的目的，因此，可以按照一种算法编写软件，利用通用计算机实现滤波；也可以按照算法选用硬件组成专用计算机实现滤波。

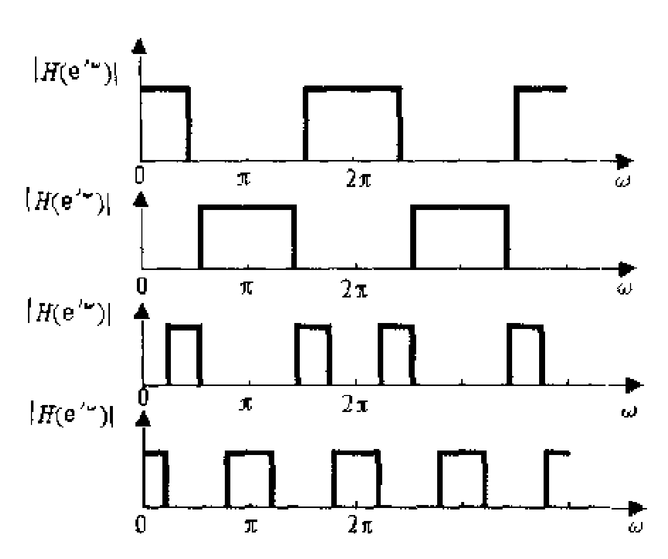


图 2-12 各种数字滤波器的理想频率特性

数字滤波器和模拟滤波器一样，从频率响应分类可以划分为低通、高通、带通和带阻等数字滤波器。由于序列的傅里叶变换具有以 2π为周期的周期性，滤波器的特性也具有这一周期性，各种数字滤波器的理想频率特性如图 2-12 所示。

数字滤波器的设计要经过三步完成：

① 确定指标　在设计一个滤波器前，必须有一些指标。这些指标要根据应用要求确定。

② 逼近　确定技术指标后，利用已有的数学和基本原理，提出一个滤波器模型来逼近给定的指标。

③ 实现　上面两步得到的结果是系统函数 $H(z)$ 或单位脉冲响应 $h(n)$ 描述的滤波器。根据这个描述用上一节所讨论的在计算机上用软件实现或以硬件实现。

幅度指标可以两种方式给出。第一种叫绝对指标，它提供对幅度响应函数$|H(e^{j\omega})|$的要求。一般用于 FIR 滤波器。第二种叫相对指标，它以分贝值（dB）的形式提出要求，一般用于 IIR 滤波器。

定义

$$\text{dB 值}=-20\lg\frac{|H(e^{j\omega})|}{|H(e^{j\omega})|_{\max}}$$

以低通滤波器为例表示绝对指标和相对指标。

绝对指标：$[0,\omega_p]$段叫做通带，δ_1 是在理想通带响应中能接受的容限（或波动）；$[\omega_r,\pi]$段叫做阻带，δ_2 是在理想阻带响应中能接受的容限（或波动）；$[\omega_p,\omega_r]$段叫做过渡带，在此段上对幅度响应没有限制。

相对指标：A_p 是通带衰减的 dB 数（或波动）；A_r 是阻带衰减的 dB 数。

因此有

$$A_p = -20\lg\frac{1-\delta_1}{1+\delta_1} > 0\,(\approx 0)$$

$$A_r = \ 20\lg\frac{\delta_2}{1+\delta_1} > 0\,(\geqslant 1)$$

2.2.1 线性相位 FIR 滤波器的特征

FIR 数字滤波器的系统函数为 $H(z)=\sum_{n=0}^{N-1}h(n)z^{-n}$

（1）线性相位条件

FIR 数字滤波器的 $h(n)$为实数，且满足以下任一条件

$$h(n)=h(N-1-n) \quad 偶对称$$

$$h(n)=-h(N-1-n) \quad 奇对称$$

其对称中心在 $n=\dfrac{N-1}{2}$ 处，则滤波器就具有准确的线性相位。

（2）线性相位特点

在以下表示中将频率响应 $H(e^{j\omega})$表示成

$$H(e^{j\omega})=H(\omega)e^{j\theta(\omega)}$$

式中 $H(\omega)$——幅度函数，它是一个纯实数，可以是正或负数；

$\theta(\omega)$——相位函数。

① $h(n)$偶对称情况

$$H(z)=\sum_{n=0}^{N-1}h(n)z^{-n}=z^{-\left(\frac{N-1}{2}\right)}\sum_{n=0}^{N-1}h(n)\frac{z^{-\left(n-\frac{N-1}{2}\right)}+z^{\left(n-\frac{N-1}{2}\right)}}{2}$$

因此

$$H(e^{j\omega})=e^{-j\left(\frac{N-1}{2}\right)\omega}\sum_{n=0}^{N-1}h(n)\cos\left[\left(\frac{N-1}{2}-n\right)\omega\right]$$

$$H(\omega)=\sum_{n=0}^{N-1}h(n)\cos\left[\left(\frac{N-1}{2}-n\right)\omega\right]$$

$$\theta(\omega)=-\left(\frac{N-1}{2}\right)\omega$$

$\theta(\omega)$是严格的线性相位，滤波器有 $\dfrac{N-1}{2}$ 个抽样的延时。

② $h(n)$奇对称情况

$$H(\omega)=\sum_{n=0}^{N-1}h(n)\sin\left[\left(\frac{N-1}{2}-n\right)\omega\right]$$

$$\theta(\omega)=-\left(\frac{N-1}{2}\right)\omega+\frac{\pi}{2}$$

$\theta(\omega)$是严格的线性相位，滤波器有 $\dfrac{N-1}{2}$ 个抽样的延时，而且产生一个 $\dfrac{\pi}{2}$ 的相移。称为正交网络。

（3）幅度函数特点

① $h(n)$偶对称，N 为奇数(1 型)

$$H(\omega)=\sum_{n=0}^{(N-1)/2}a(n)\cos(\omega n)$$

其中

$$a(0)=h\left(\frac{N-1}{2}\right)$$

$$a(n)=2h\left(\frac{N}{2}-n\right)\quad n=1,2,\cdots,\frac{N-1}{2}$$

$H(\omega)$对$\omega=0, \pi, 2\pi$呈偶对称，在 MATLAB 中用 hr_type1 函数实现。

［例 1］ 设$h(n)=\{-4,2,-1,-2,5,6,5,-2,-1,2,-4\}$，求振幅响应$H(\omega)$及零极点分布。

```
%s151.m
h=[-4,2,-1,-2,5,6,5,-2,-1,2,-4];
M=length(h);n=0:M-1;
[Hr,w,a,L]=hr_type1(h);
subplot(221);stem(n,h);title('1 型单位脉冲响应 h(n)');
subplot(222);plot(w/pi,Hr);title('1 型振幅响应 H(w)');
subplot(223);stem(0:L,a);title('a(n)系数');
subplot(224);zplane(h,1);title('零极点分布');
```

结果如图 2-13 所示。

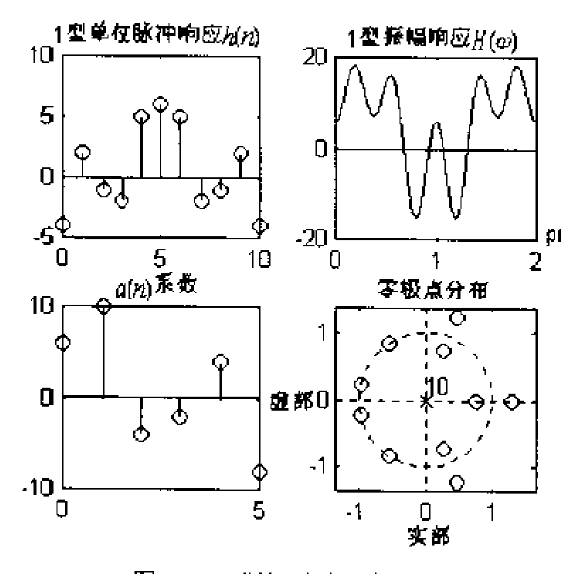


图 2-13 振幅响应（例 1）

由于$N=11$，$h(n)$偶对称，故振幅响应$H(\omega)$对$\omega=0, \pi, 2\pi$呈偶对称。在z平面上有 10 个零点，这些零点有一定的对称性；在原点有一个 10 重极点。

```
function[Hr,w,a,L]=hr_type1(h);
%计算 1 型滤波器设计的振幅响应
%Hr=振幅响应
%a=1 型滤波器的系数
%L=Hr 的阶次
%h=1 型滤波器的单位脉冲响应
M=length(h);L=(M-1)/2;
a=[h(L+1) 2*h(L:-1:1)];
n=[0:1:L];
w=[0:1:500]'*2*pi/500;
Hr=cos(w*n)*a';
```

② $h(n)$偶对称，N为偶数(2 型)

$$H(\omega)=\sum_{n=1}^{N/2}b(n)\cos\left[\omega\left(n-\frac{1}{2}\right)\right]$$

其中

$$b(n)=2h\left(\frac{N}{2}-n\right)\quad n=1,2,\cdots,\frac{N}{2}$$

$H(\omega)$对$\omega=\pi$呈奇对称，$H(\pi)=0$ 故高通滤波器不能用这种形式，在 MATLAB 中用 hr_type2 函数实现。

［例 2］ 设$h(n)=\{-4,1,-1,-2,5,6,6,5,-2,-1,1,-4\}$求振幅响应$H(\omega)$及零极点分布。

```
%s152.m
```

```
h=[-4,1,-1,-2,5,6,6,5,-2,-1,1,-4];
M=length(h);n=0:M-1;
[Hr,w,b,L]=hr_type2(h);
subplot(221);stem(n,h);title('2 型单位脉冲响应 h(n)');
subplot(222);plot(w/pi,Hr);title('2 型振幅响应 H(w)');
subplot(223);stem(1:L,b);title('b(n)系数');
subplot(224);zplane(h,1);title('零极点分布');
```

结果如图 2-14 所示。

由于N=12，$h(n)$偶对称，故振幅响应$H(\omega)$对$\omega=\pi$呈奇对称。在 z 平面上有 11 个零点，这些零点有一定的对称性：在原点有一个 11 重极点。

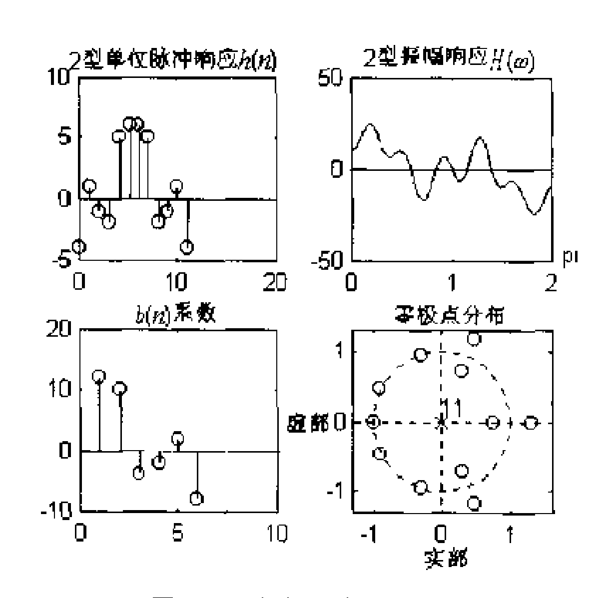


图 2-14 振幅响应（例 2）

```
function[Hr,w,b,L]=hr_type2(h)
%计算 2 型滤波器设计的振幅响应
%Hr=振幅响应
%b=2 型滤波器的系数
%L=Hr 的阶次
%h=2 型滤波器的单位脉冲响应
M=length(h);
L=M/2;b=2*[h(L:-1:1)];
n=[1:1:L];n=n-0.5;
w=[0:1:500]'*2*pi/500;
Hr=cos(w*n)*b';
```

③ $h(n)$奇对称，N 为奇数(3 型)

$$H(\omega)=\sum_{n=1}^{(N-3)/2}c(n)\sin(\omega n)$$

其中

$$c(n)=2h\left(\frac{N-1}{2}-n\right)\quad n=1,2,\cdots,\frac{N-1}{2}$$

$H(\omega)$对$\omega=0,\ \pi,\ 2\pi$呈奇对称，$\omega=0,\ \pi,\ 2\pi$时 $H(\omega)=0$ 故低、高通滤波器不能用这种形式，在 MATLAB 中用 hr_type3 函数实现。

［例 3］ 设$h(n)=\{-4,1,-1,-2,5,0,-5,2,1,-1,4\}$，求振幅响应 $H(\omega)$及零极点分布。

```
%s153.m
h=[-4,1,-1,-2,5,0,-5,2,1,-1,4];
M=length(h);n=0:M-1;
[Hr,w,c,L]=hr_type3(h);
subplot(221);stem(n,h);title('3 型单位脉冲响应 h(n)');
subplot(222);plot(w/pi,Hr);title('3 型振幅响应 H(w)');
subplot(223);stem(0:L,c);title('c(n)系数');
subplot(224);zplane(h,1);title('零极点分布');
```

结果如图 2-15 所示。

由于 N =11，$h(n)$奇对称，故振幅响应 $H(\omega)$对$\omega=0,\ \pi,\ 2\pi$呈奇对称。在 z 平面上有 10 个零点，这些零点有一定的对称性：在原点有一个 10 重极点。

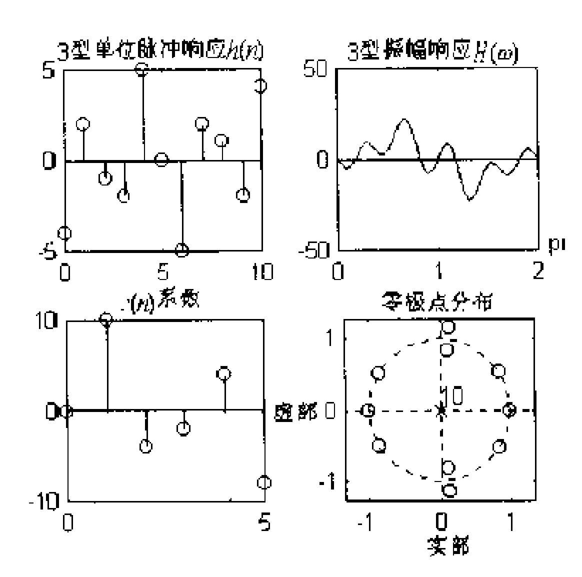


图 2-15　振幅响应（例 3）

```
function[Hr,w,c,L]=hr_type3(h);
%计算 3 型滤波器设计的振幅响应
%Hr=振幅响应
%b=3 型滤波器的系数
%L=Hr 的阶次
%h=3 型滤波器的单位脉冲响应
M=length(h);
L=(M-1)/2;
c=[2*h(L+1:-1:1)];
n=[0:1:L];
w=[0:1:500]'*2*pi/500;
Hr=sin(w*n)*c';
```

④ $h(n)$奇对称，N 为偶数(4 型)

$$H(\omega)=\sum_{n=1}^{N/2}d(n)\sin\left[\omega\left(n-\frac{1}{2}\right)\right]$$

其中

$$d(n)=2h\left(\frac{N}{2}-n\right)\quad n=1,2,\cdots,\frac{N}{2}$$

$H(\omega)$对$\omega=0$, 2π呈奇对称，对$\omega=\pi$呈偶对称，$\omega=0$, 2π时 $H(\omega)=0$ 故低通滤波器不能用这种形式，在 MATLAB 中用 hr_type4 函数实现。

[例 4]　设 $h(n)=\{-4,2,-1,-2,5,-6,6,-5,2,1,-2,4\}$，求振幅响应 $H(\omega)$及零极点分布。

```
%s154.m
h=[-4,2,-1,-2,5,-6,6,-5,2,1,-2,4];
M=length(h);n=0:M-1;
[Hr,w,d,L]=hr_type4(h);
subplot(221);stem(n,h);title('4 型单位脉冲响应 h(n)');
subplot(222);plot(w/pi,Hr);title('4 型振幅响应 H(w)');
subplot(223);stem(1:L,d);title('d(n)系数');
subplot(224);zplane(h,1);title('零极点分布');
```

结果见图 2-16。

由于 $N=12$，$h(n)$偶对称，故振幅响应$H(\omega)$对$\omega=0$, 2π呈奇对称，对$\omega=\pi$呈偶对称，在 z 平面上有 11 个零点，这些零点有一定的对称性；在原点有一个 11 重极点。

```
function[Hr,w,d,L]=hr_type4(h);
%计算 4 型滤波器设计的振幅响应
%Hr=4 型振幅响应
%b=4 型滤波器的系数
%L=Hr 的阶次
%h=4 型滤波器的单位脉冲响应
M=length(h);
L=M/2;
d=2*[h(L:-1:1)];
```

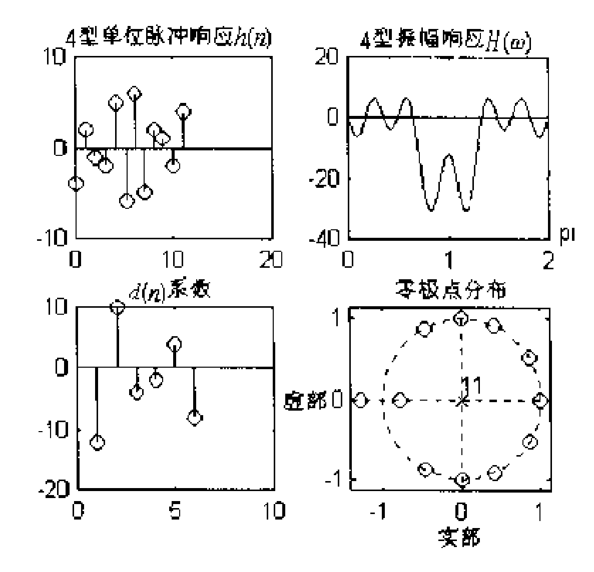


图 2-16　振幅响应（例 4）

```
n=[1:1:L];n=n-0.5;
w=[0:1:500]'*2*pi/500;
Hr=sin(w*n)*d';
```

（4）零点位置

线性相位 FIR 滤波器的系统函数满足 $H(z)=\pm z^{-(N-1)}H(z^{-1})$

零点位置有四种可能：

① 零点 z_i 既不在实轴上，也不在单位圆上，是互为倒数的两组共轭对。

② 零点 z_i 不在实轴上，但在单位圆上，是共轭对。

③ 零点 z_i 在实轴上，不在单位圆上，是实数零点和倒数零点。

④ 零点 z_i 既在实轴上，也在单位圆上，此时只有一个零点 $z=1$ 或 $z=-1$。

2.2.2 利用窗函数技术设计

设所要求的理想数字滤波器的频率响应为 $H_d(e^{j\omega})$，$h_d(n)$ 是与其对应的单位脉冲响应，因此

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega$$

由于 $H_d(e^{j\omega})$ 是矩形频率特性，故 $h_d(n)$ 一定是无限长的非因果序列。

而所要设计的是 FIR 数字滤波器，其单位脉冲响应 $h(n)$ 必然是有限长的，所以要用有限的 $h(n)$ 来逼近无限长的 $h_d(n)$，最有效的方法是截断 $h_d(n)$，即用有限长的窗函数 $w(n)$ 来截取 $h_d(n)$，表示为

$$h(n)=h_d(n)w(n)$$

这种设计方法称为窗函数设计法。

2.2.2.1 数字低通滤波器的窗函数设计

理想低通数字滤波器的频率响应 $H_d(e^{j\omega})$ 为

$$H_d(e^{j\omega})=\begin{cases}1\cdot e^{-j\omega\alpha}, & |\omega|\leqslant\omega_c\\ 0, & \pi\geqslant|\omega|>\omega_c\end{cases}$$

式中　ω_c——截止频率，rad；

　　　α——采样延迟。

则理想数字低通滤波器的单位脉冲响应 $h_d(n)$ 为

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega=\frac{1}{2\pi}\int_{-\omega_c}^{\omega_c}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega=\frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}$$

$h_d(n)$ 为无限长非因果序列，关于 α 偶对称。

为了从 $h_d(n)$ 得到一个 FIR 数字滤波器，必须同时在两边截取 $h_d(n)$，要得到一个因果的线性相位 FIR 滤波器，它的 $h(n)$ 的长度为 N，必须有

$$h(n)=\begin{cases}h_d(n), & 0\leqslant n\leqslant N-1\\ 0, & \text{其他}\end{cases}$$

这种截取可看作是

$$\alpha=\frac{N-1}{2}$$

$$h(n)=h_d(n)w(n)$$

其中

$$w(n)=\begin{cases}1, & 0\leqslant n\leqslant N-1\\ 0, & 其他\end{cases}\quad 矩形窗$$

$h(n)$为关于α偶对称的有限长因果序列。N为奇数时是1型；N为偶数时是2型。在MATLAB中利用ideal_lp函数，计算理想低通滤波器的单位脉冲响应$h_d(n)$。

```
function hd=ideal_lp(wc,N);
alpha=(N-1)/2;
n=0:1:N-1;
m=n-alpha+eps;
hd=sin(wc*m)./(pi*m);
```

在频域中，FIR数字滤波器的频率响应$H(e^{j\omega})$为：

$$H(e^{j\omega})=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\theta})W(e^{j(\omega-\theta)})d\theta$$

因而$H(e^{j\omega})$逼近$H_d(e^{j\omega})$的好坏，完全取决于窗函数的频率特性$W(e^{j\omega})$。

在MATLAB中利用freqz_m计算绝对和相对的幅度响应。

```
function[db,mag,pha,w]=freqz_m(b,a);
[H,w]=freqz(b,a,1000,'whole');
H=(H(1:1:501))';
w=(w(1:1:501))';
mag=abs(H);
db=20*log10((mag+eps)/max(mag));
pha=angle(H);
```

［例5］ 根据下列技术指标，设计一个数字FIR低通滤波器。

$$\omega_p=0.2\pi, A_p=0.25\text{dB}$$

$$\omega_r=0.4\pi, A_r=50\text{dB}$$

选择一个恰当的窗函数，确定单位脉冲响应，绘出所设计的滤波器的幅度响应。

解：根据窗函数最小阻带衰减的特性，只有海明窗和布莱克曼窗可提供大于50dB的衰减。故选择海明窗，它提供较小的过渡带，因此具有较小的阶数。

```
%s155.m
wp=0.2*pi;wr=0.4*pi;
tr_width=wr-wp;%过渡带宽度
N=ceil(6.6*pi/tr_width)+1  %滤波器的长度,N=奇数为1型; N=偶数为2型
n=0:1:N-1;
wc=(wr+wp)/2;%理想低通滤波器的截止频率
hd=ideal_lp(wc,N);%理想低通滤波器的单位脉冲响应
w_ham=(hamming(N))';%海明窗
h=hd.*w_ham;%截取得到实际单位脉冲响应
[db,mag,pha,w]=freqz_m(h,[1]);%计算实际滤波器的幅度响应
delta_w=2*pi/1000;
Ap=-(min(db(1:1:wp/delta_w+1)));%实际通带波动
Ar=-round(max(db(wr/delta_w+1:1:501)));%最小阻带衰减
subplot(221);stem(n,hd);title('理想单位脉冲响应hd(n)')
subplot(222);stem(n,w_ham);title('海明窗w(n)')
```

```
subplot(223);stem(n,h);title('实际单位脉冲响应h(n)')
subplot(224);plot(w/pi,db);title('幅度响应(dB)')
axis([0,1,-100,10]);
```

结果见图 2-17。

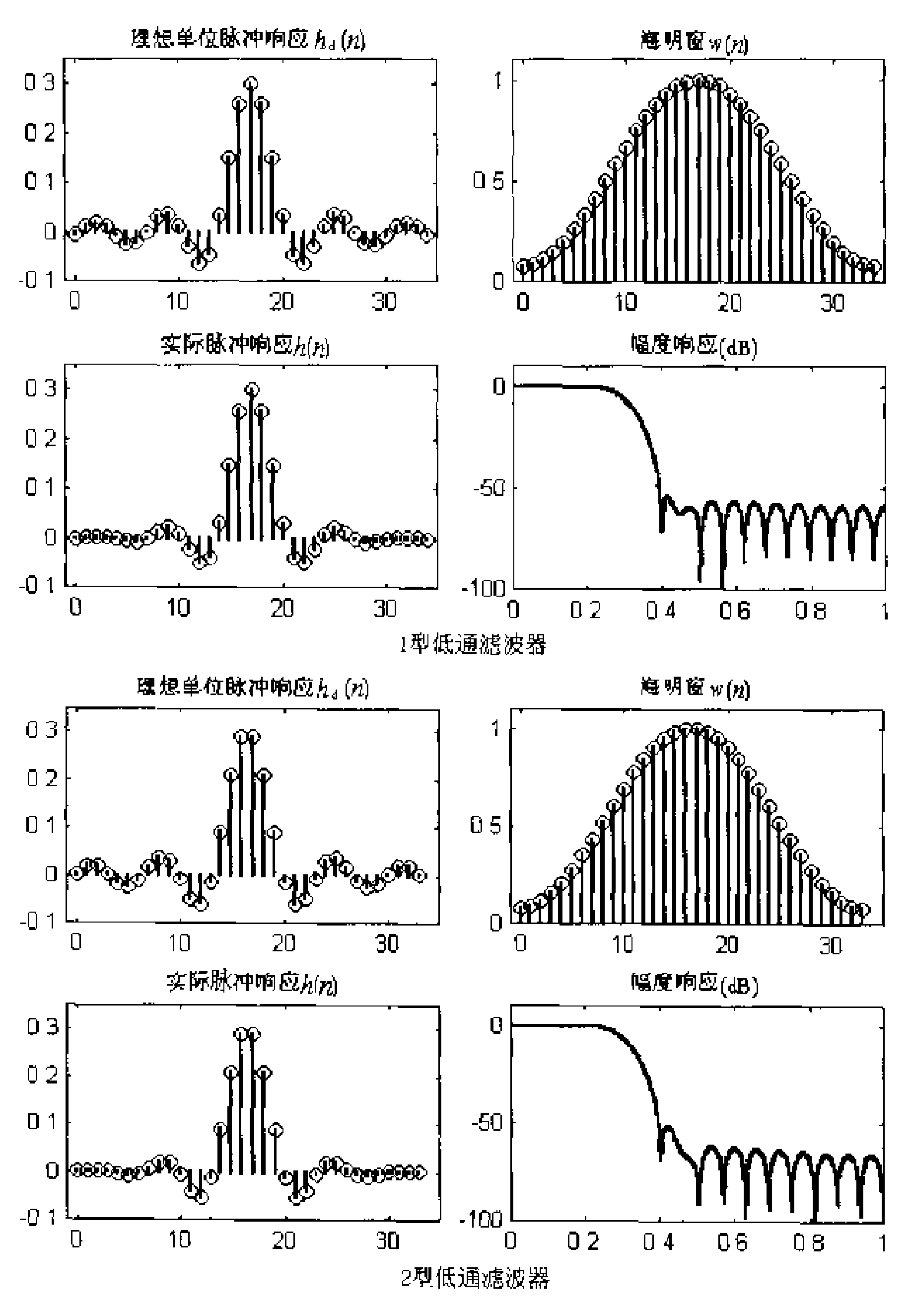


图 2-17　滤波器的幅度响应

滤波器的长度为 35，实际通带波动 0.0301dB ，最小阻带衰减 55dB，满足设计要求。
滤波器的长度为 34，实际通带波动 0.0477dB ，最小阻带衰减 52dB，满足设计要求。

2.2.2.2　数字高通滤波器的窗函数设计

① 理想高通数字滤波器的频率响应 $H_d(e^{j\omega})$ 为

$$H_d(e^{j\omega})=\begin{cases}1\cdot e^{-j\omega\alpha}, & \omega_c\leqslant|\omega|\leqslant\pi\\ 0, & |\omega|<\omega_c\end{cases}$$

式中 ω_c——截止频率，rad；

α——采样延迟。

则理想高通数字滤波器的单位脉冲响应 $h_d(n)$为

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega=\frac{1}{2\pi}\left[\int_{-\pi}^{-\omega_c}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega+\int_{\omega_c}^{\pi}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega\right]$$
$$=\frac{\sin[\pi(n-\alpha)-\omega_c(n-\alpha)]}{\pi(n-\alpha)}$$

$h_d(n)$为无限长非因果序列，关于α偶对称，$\alpha=\dfrac{N-1}{2}$。

数字高通滤波器的单位脉冲响应为 $h(n)=h_d(n)w(n)$，偶对称因果序列，应使 N 为奇数（1 型）。利用 ideal_hp 函数，计算理想高通滤波器的单位脉冲响应 $h_d(n)$。

```
function hd=ideal_hp(wc,N);
alpha=(N-1)/2;
n=0:1:N-1;
m=alpha+eps;
hd=[sin(pi*m)-sin(wc*m)]./(pi*m);
```

[例 6] 根据下列技术指标，设计一个数字 FIR 高通滤波器。

$$\omega_p=0.6\pi, A_p=0.25\text{dB}$$
$$\omega_r=0.4\pi, A_r=40\text{dB}$$

选择一个恰当的窗函数，确定单位脉冲响应，绘出所设计的滤波器的幅度响应。

解：根据窗函数最小阻带衰减的特性，选择汉宁窗可以达到 44dB 的最小阻带衰减，它提供的过渡带宽为 $6.2\pi/N$，因此具有较小的阶数。

```
%s156.m
wp=0.6*pi;wr=0.4*pi;
tr_width=wp-wr;
N=ceil(6.2*pi/tr_width) %应使截取的长度为奇数
n=0:1:N-1;
wc=(wr+wp)/2;%理想高通滤波器的截止频率
hd=ideal_hp(wc,N);%1 型理想高通滤波器响应
w_ham=(hanning(N))';
h=hd.*w_ham;
[db,mag,pha,w]=freqz_m(h,[1]);
delta_w=2*pi/1000;
Ap=-(min(db(wp/delta_w+1:1:501))) %实际通带波动
Ar=-round(max(db(1:1:wr/delta_w+1))) %最小阻带衰减
subplot(221);stem(n,hd);title('理想单位脉冲响应 hd(n)');
subplot(222);stem(n,w_ham);title('汉宁窗 w(n)')
subplot(223);stem(n,h);title('实际单位脉冲响应 h(n)')
subplot(224);plot(w/pi,db);title('幅度响应(dB)')
```

结果见图 2-18。

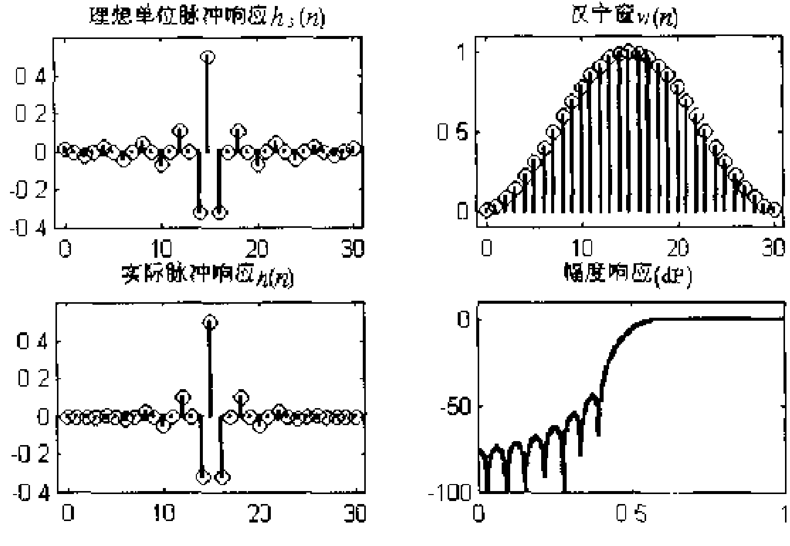


图 2-18　1 型高通滤波器的幅度响应

滤波器的长度为 31，实际通带波动 0.0887dB，最小阻带衰减 44dB，满足设计要求。

② 理想高通数字滤波器的频率响应 $H_d(e^{j\omega})$为

$$H_d(e^{j\omega})=\begin{cases}1\cdot e^{-j\left(\omega\alpha-\frac{\pi}{2}\right)}, & \omega_c\leqslant\omega\leqslant\pi\\ 1\cdot e^{-j\left(\omega\alpha+\frac{\pi}{2}\right)}, & -\pi\leqslant\omega\leqslant-\omega_c\\ 0, & |\omega|<\omega_c\end{cases}$$

式中　ω_c——截止频率，rad；

　　α——采样延迟。

则理想高通数字滤波器的单位脉冲响应 $h_d(n)$为

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega=\frac{1}{2\pi}\left[\int_{-\pi}^{-\omega_c}1\cdot e^{-j\left(\omega\alpha+\frac{\pi}{2}\right)}e^{j\omega n}d\omega+\int_{\omega_c}^{\pi}1\cdot e^{-j\left(\omega\alpha-\frac{\pi}{2}\right)}e^{j\omega n}d\omega\right]$$

$$=\frac{\cos[\pi(n-\alpha)]-\omega_c(n-\alpha)]}{\pi(n-\alpha)}$$

$h_d(n)$为无限长非因果序列，关于α 奇对称，$\alpha=\dfrac{N-1}{2}$。

数字高通滤波器的单位脉冲响应为 $h(n)=h_d(n)w(n)$，奇对称因果序列，应使 N 为偶数（4 型）。利用 ideal_hp1 函数，计算理想高通滤波器的单位脉冲响应 $h_d(n)$。

```
function hd=ideal_hp1(wc,N);
alpha=(N-1)/2;
n=[0:1:(N-1)];
m=n-alpha+eps;
hd=[cos(pi*m)-cos(wc*m)]./(pi*m);
```

[例 7]　根据下列技术指标，设计一个具有$\dfrac{\pi}{2}$相移的数字 FIR 高通滤波器。

$$\omega_p=0.6\pi, A_p=0.25\text{dB}$$

$$\omega_r=0.4\pi, A_r=40\text{dB}$$

选择一个恰当的窗函数，确定单位脉冲响应，绘出所设计的滤波器的幅度响应。

解： 根据窗函数最小阻带衰减的特性，选择汉宁窗可以达到 44dB 的最小阻带衰减，它提供的过渡带宽为 $6.2\pi/N$，因此具有较小的阶数。

```
%s157.m
wp=0.6*pi;wr=0.4*pi;
tr_width=wp-wr;
N=ceil(6.2*pi/tr_width)+1 %应使截取的长度为偶数
n=0:1:N-1;
wc=(wr+wp)/2;%理想高通滤波器的截止频率
hd=ideal_hp1(wc,N);%4 型理想高通滤波器响应
w_ham=(hanning(N))';
h=hd.*w_ham;
[db,mag,pha,w]=freqz_m(h,[1]);
delta_w=2*pi/1000;
Ap=-(min(db(wp/delta_w+1:1:501))) %实际通带波动
Ar=-round(max(db(1:1:wr/delta_w+1))) %最小阻带衰减
subplot(221);stem(n,hd);title('理想单位脉冲响应 hd(n)');
subplot(222);stem(n,w_ham);title('汉宁窗 w(n)')
subplot(223);stem(n,h);title('实际单位脉冲响应 h(n)')
subplot(224);plot(w/pi,db);title('幅度响应(dB)')
```

结果见图 2-19。

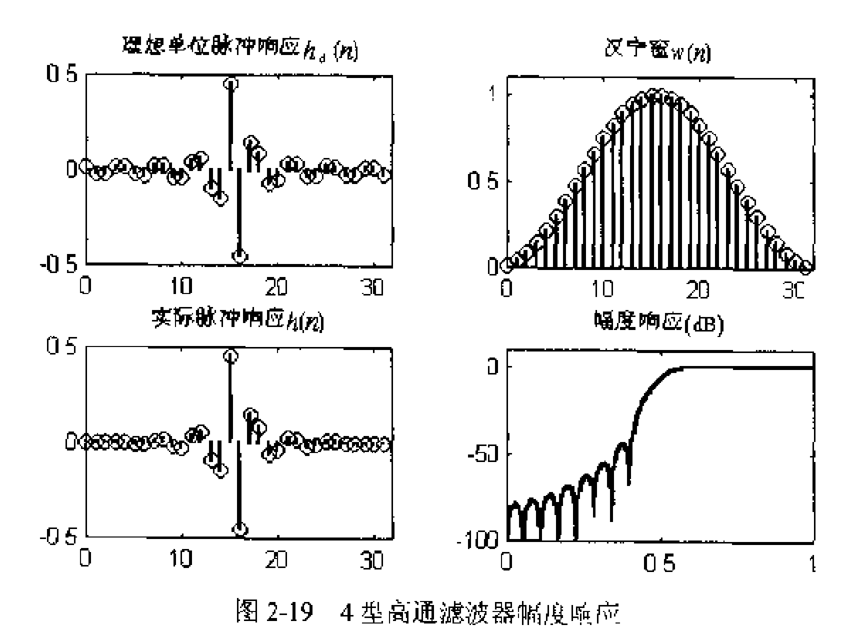


图 2-19 4 型高通滤波器幅度响应

滤波器的长度为 32，实际通带波动 0.0771dB，最小阻带衰减 44dB，满足设计要求。

2.2.2.3 数字带通滤波器的窗函数设计

① 理想带通数字滤波器的频率响应 $H_d(e^{j\omega})$ 为

$$H_d(e^{j\omega})=\begin{cases}1\cdot e^{-j\omega\alpha}, & \omega_{c1}\leqslant\omega\leqslant\omega_{c2}\\ 0, & |\omega|<\omega_{c1},\ |\omega|>\omega_{c2}\end{cases}$$

式中 ω_{c1} —— 下截止频率，rad;

ω_{c2} ——上截止频率，rad；

α —— 采样延迟。

则理想带通数字滤波器的单位脉冲响应 $h_d(n)$为

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega=\frac{1}{2\pi}\left[\int_{-\omega_{c2}}^{-\omega_{c1}}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega+\int_{\omega_{c1}}^{\omega_{c2}}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega\right]$$

$$=\frac{\sin[\omega_{c2}(n-\alpha)-\omega_{c1}(n-\alpha)]}{\pi(n-\alpha)}$$

$h_d(n)$为无限长非因果序列，关于α 偶对称，$\alpha=\frac{N-1}{2}$。

数字带通滤波器的单位脉冲响应为 $h(n)=h_d(n)w(n)$，偶对称因果序列。当 N 为奇数是 1 型，当 N 为偶数是 2 型。此时利用 ideal_bp1 函数，计算理想带通滤波器的单位脉冲响应 $h_d(n)$。

```
function hd=ideal_bp1(wc1,wc2,N);
alpha=(N-1)/2;
n=[0:1:(N-1)];
m=n-alpha+eps;
hd=[sin(wc2*m)-sin(wc1*m)]./(pi*m);
```

[例 8] 根据下列技术指标，设计一个数字 FIR 带通滤波器。

低端阻带边缘	$\omega_{r1}=0.2\pi, A_{r1}=60\text{dB}$
低端通带边缘	$\omega_{p1}=0.4\pi, A_{p1}=1\text{dB}$
高端通带边缘	$\omega_{p2}=0.6\pi, A_{p2}=1\text{dB}$
高端阻带边缘	$\omega_{r2}=0.8\pi, A_{r2}=60\text{dB}$

选择一个恰当的窗函数，确定单位脉冲响应，绘出所设计的滤波器的幅度响应。

解：根据窗函数最小阻带衰减的特性，选择布莱克曼窗可以达到 75dB 的最小阻带衰减，它提供的过渡带宽为 $11\pi/N$。

```
%s158.m
wr1=0.2*pi;wp1=0.4*pi;
wp2=0.6*pi;wr2=0.8*pi;
tr_width=min((wp1-wr1),(wr2-wp2))
N=ceil(11*pi/tr_width)+1 %长度为奇数 1 型，长度为偶数 2 型
n=0:1:N-1;
wc1=(wr1+wp1)/2;wc2=(wr2+wp2)/2;%理想带通滤波器的上下截止频率
hd=ideal_bp1(wc1,wc2,N);%1、2 型理想带通滤波器响应
w_ham=(blackman(N))';
h=hd.*w_ham;
[db,mag,pha,w]=freqz_m(h,[1]);
delta_w=2*pi/1000;
Ap=-(min(db(wp1/delta_w+1:1:wp2/delta_w+1))) %实际通带波动
Ar=-round(max(db(wr2/delta_w+1:1:501))) %最小阻带衰减
subplot(221);stem(n,hd);title('理想单位脉冲响应 hd(n)');
subplot(222);stem(n,w_ham);title('布莱克曼窗 w(n)');
subplot(223);stem(n,h);title('实际单位脉冲响应 h(n)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)')
```

结果见图2-20。

滤波器的长度为56，实际通带波动0.0027dB，最小阻带衰减73dB，满足设计要求。

滤波器的长度为55，实际通带波动0.0034dB，最小阻带衰减71dB，满足设计要求。

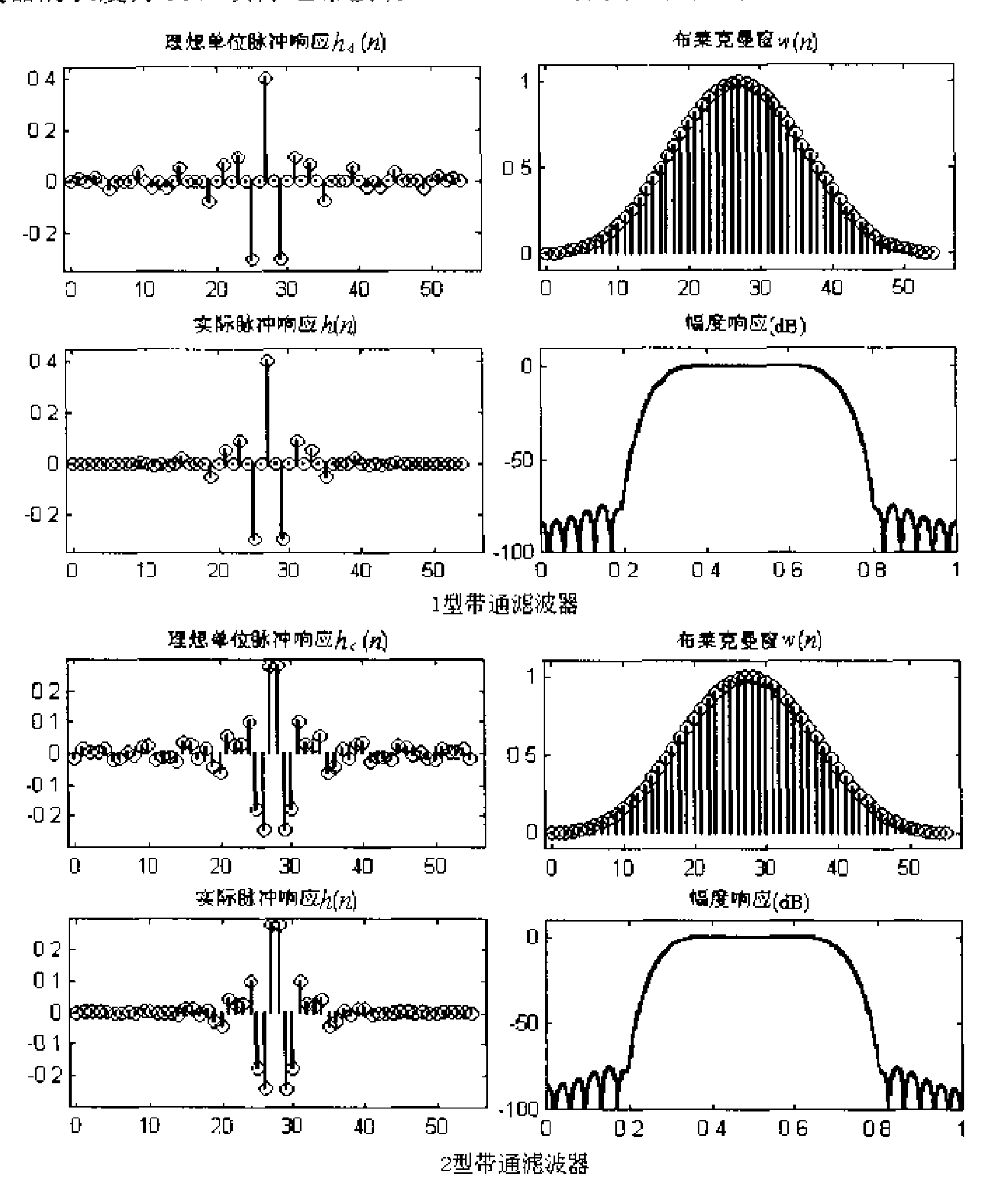


图2-20 带通滤波器幅度响应

② 理想带通数字滤波器的频率响应$H_d(e^{j\omega})$为

$$H_d(e^{j\omega})=\begin{cases}1\cdot e^{-j\left(\omega\alpha-\frac{\pi}{2}\right)}, & \omega_{c1}\leqslant\omega\leqslant\omega_{c2}\\ 1\cdot e^{-j\left(\omega\alpha+\frac{\pi}{2}\right)}, & \omega_{c2}\leqslant\omega\leqslant\omega_{c1}\\ 0, & |\omega|<\omega_{c1},\ |\omega|>\omega_{c2}\end{cases}$$

式中 ω_{c1}——下截止频率，rad；

ω_{c2}——上截止频率，rad；

α——采样延迟。

则理想带通数字滤波器的单位脉冲响应$h_d(n)$为

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega=\frac{1}{2\pi}\left[\int_{-\omega_{c2}}^{-\omega_{c1}}1\cdot e^{-j\left(\omega\alpha+\frac{\pi}{2}\right)}\cdot e^{j\omega n}d\omega+\int_{\omega_c}^{\omega_{c2}}1\cdot e^{j\left(\omega\alpha-\frac{\pi}{2}\right)}\cdot e^{j\omega n}d\omega\right]$$

$$=\frac{\cos[\omega_{c2}(n-\alpha)-\omega_{c1}(n-\alpha)]}{\pi(n-\alpha)}$$

$h_d(n)$为无限长非因果序列，关于α 奇对称，$\alpha=\frac{N-1}{2}$。

数字带通滤波器的单位脉冲响应为 $h(n)=h_d(n)w(n)$，奇对称因果序列。当 N 为奇数是 3 型，当 N 为偶数是 4 型。此时利用 ideal_bp2 函数，计算理想带通滤波器的单位脉冲响应 $h_d(n)$。

```
function hd=ideal_bp2(wc1,wc2,N);
alpha=(N-1)/2;
n=0:1:N-1;
m=n-alpha+eps;
hd=[cos(wc2*m)-cos(wc1*m)]./(pi*m);
```

［例 9］ 根据下列技术指标，设计一个具有$\frac{\pi}{2}$ 相移的数字 FIR 带通滤波器。

低端阻带边缘	$\omega_{r1}=0.2\pi, A_{r1}=20\text{dB}$
低端通带边缘	$\omega_{p1}=0.4\pi, A_{p1}=1\text{dB}$
高端通带边缘	$\omega_{p2}=0.6\pi, A_{p2}=1\text{dB}$
高端阻带边缘	$\omega_{r2}=0.8\pi, A_{r2}=20\text{dB}$

选择一个恰当的窗函数，确定单位脉冲响应，绘出所设计的滤波器的幅度响应。

解：根据窗函数最小阻带衰减的特性，选择三角窗可以达到 25dB 的最小阻带衰减，它提供的过渡带宽为 $6.1\pi/N$。

```
%s159.m
wr1=0.2*pi;wp1=0.4*pi;wp2=0.6*pi;wr2=0.8*pi;
tr_width=min((wp1-wr1),(wr2-wp2))
N=ceil(6.1*pi/tr_width) %长度为奇数 3 型，长度为偶数 4 型
n=0:1:N-1;
wc1=(wr1+wp1)/2;wc2=(wr2+wp2)/2;%理想带通滤波器的上下截止频率
hd=ideal_bp2(wc1,wc2,N);%3、4 型理想带通滤波器响应
w_ham=(triang(N))';
h=hd.*w_ham;
[db,mag,pha,w]=freqz_m(h,[1]);
delta_w=2*pi/1000;
Ap=-(min(db(wp1/delta_w+1:1:wp2/delta_w+1))) %实际通带波动
Ar=-round(max(db(wr2/delta_w+1:1:501))) %最小阻带衰减
subplot(221);stem(n,hd);title('理想单位脉冲响应 hd(n)');
subplot(222);stem(n,w_ham);title('三角窗 w(n)');
subplot(223);stem(n,h);title('实际单位脉冲响应 h(n)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)')
```

结果见图 2-21 及图 2-22。

滤波器的长度为 31，实际通带波动 0.1241dB，最小阻带衰减 29dB，满足设计要求。

滤波器的长度为 32，实际通带波动 0.1331dB，最小阻带衰减 27dB，满足设计要求。

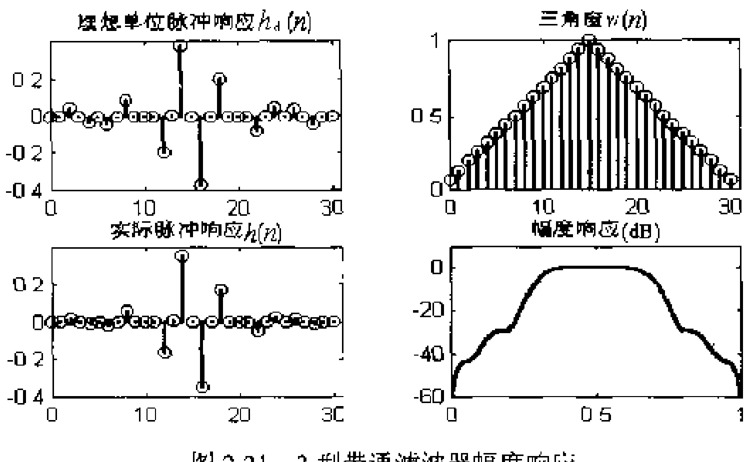


图 2-21 3 型带通滤波器幅度响应

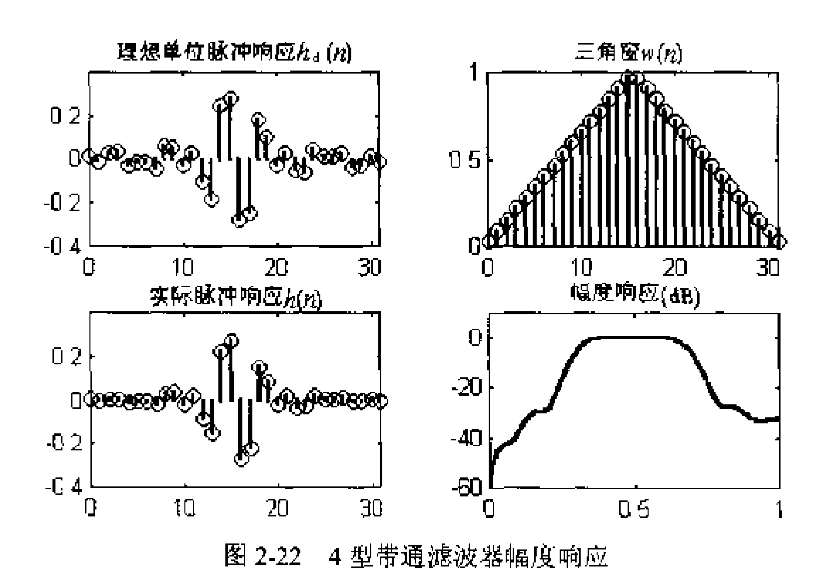


图 2-22 4 型带通滤波器幅度响应

2.2.2.4 数字带阻滤波器的窗函数设计

理想带阻数字滤波器的频率响应 $H_d(e^{j\omega})$为

$$H_d(e^{j\omega})=\begin{cases}1\cdot e^{-j\omega\alpha}, & |\omega|\leqslant\omega_{c1},\ \pi\geqslant|\omega|\geqslant\omega_{c2}\\ 0, & \omega_{c1}<|\omega|<\omega_{c2}\end{cases}$$

式中 ω_{c1}——下截止频率，rad；

ω_{c2}——上截止频率，rad；

α——采样延迟。

则理想带阻数字滤波器的单位脉冲响应 $h_d(n)$为

$$\begin{aligned}h_d(n)&=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_c(e^{j\omega})e^{j\omega n}d\omega\\&=\frac{1}{2\pi}\left[\int_{-\pi}^{-\omega_{c2}}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega+\int_{-\omega_{c1}}^{\omega_{c1}}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega+\int_{\omega_{c2}}^{\pi}1\cdot e^{-j\omega\alpha}\cdot e^{j\omega n}d\omega\right]\\&=\frac{\sin[\omega_{c1}(n-\alpha)+\pi(n-\alpha)-\omega_{c2}(n-\alpha)]}{\pi(n-\alpha)}\end{aligned}$$

$h_d(n)$为无限长非因果序列，关于α偶对称，$\alpha=\dfrac{N-1}{2}$。

数字带阻滤波器的单位脉冲响应为 $h(n)=h_d(n)w(n)$，偶对称因果序列。应当使 N 为奇数（1 型）。利用 ideal_bs 函数，计算理想带阻滤波器的单位脉冲响应 $h_d(n)$。

```
function hd=ideal_bs(wc1,wc2,N);
alpha=(N-1)/2;
n=[0:1:(N-1)];
m=n-alpha+eps;
hd=[sin(wc1*m)+sin(pi*m)-sin(wc2*m)]./(pi*m);
```

［例 10］ 根据下列技术指标，设计一个数字 FIR 带阻滤波器。

低端通带边缘 $\omega_{p1}=0.2\pi, A_{p1}=1\text{dB}$

低端阻带边缘 $\omega_{r1}=0.4\pi, A_{r1}=40\text{dB}$

高端阻带边缘 $\omega_{r2}=0.6\pi, A_{r2}=40\text{dB}$

高端通带边缘 $\omega_{p2}=0.8\pi, A_{p2}=1\text{dB}$

选择一个恰当的窗函数，确定单位脉冲响应，绘出所设计的滤波器的幅度响应。

解：根据窗函数最小阻带衰减的特性，选择汉宁窗可以达到 44dB 的最小阻带衰减，它提供的过渡带宽为 $6.2\pi/N$。

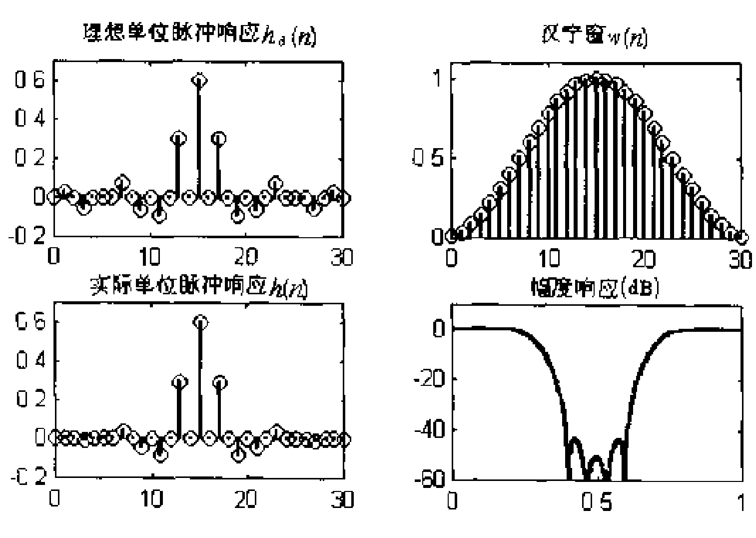


图 2-23　带阻滤波器幅度响应

```
%s160.m
wp1=0.2*pi;wr1=0.4*pi;
wp2=0.8*pi;wr2=0.6*pi;
tr_width=min((wr1-wp1),(wp2-wr2))
N=ceil(6.2*pi/tr_width)
n=0:1:N-1;
wc1=(wr1+wp1)/2;wc2=(wr2+wp2)/2;
hd=ideal_bs(wc1,wc2,N);
w_ham=(hanning(N))';
h=hd.*w_ham;
[db,mag,pha,w]=freqz_m(h,[1]);
delta_w=2*pi/1000;
Ap=-(min(db(1:1:wp1/delta_w+1)))
```

```
Ar=-round(max(db(wr1/delta_w+1:1:wr2/delta_w+1)))
subplot(221);stem(n,hd);title('理想单位脉冲响应 hd(n)')
subplot(222);stem(n,w_ham);title('汉宁窗 w(n)')
subplot(223);stem(n,h);title('实际单位脉冲响应 h(n)')
subplot(224);plot(w/pi,db);title('幅度响应(dB)')
```

结果见图 2-23。

滤波器的长度为 31，实际通带波动 0.0885dB，最小阻带衰减 44dB，满足设计要求。

2.2.3　利用频率采样技术设计

频率采样设计法是从频域出发，把给定的理想频率响应 $H_d(e^{j\omega})$等间隔采样 N 个点得到

$$H_d(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k}=H_d(k)=H(k)$$

再对 $H_d(k)$作 IDFT，得到 $h(n)$

$$h(n)=\frac{1}{N}\sum_{k=0}^{N-1}H(k)e^{j\frac{2\pi}{N}kn}\quad k=0,1,\cdots,N-1$$

将 $h(n)$作为所设计的滤波器的单位脉冲响应。

（1）为保证具有线性相位，对频率采样值的约束条件

将采样值用幅值 H_k 与相角 θ_k 表示成 $H(k)=H_k e^{j\theta_k}$

$h(n)$偶对称 N 为奇数时，约束条件为

$$H_k=H_{N-k}$$

$$\theta_k=-k\pi\left(1-\frac{1}{N}\right)$$

$h(n)$偶对称 N 为偶数时，约束条件为

$$H_k=-H_{N-k}$$

$$\theta_k=-k\pi\left(1-\frac{1}{N}\right)$$

$h(n)$奇对称 N 为奇数时，约束条件为

$$H_k=-H_{N-k}$$

$$\theta_k=-k\pi\left(1-\frac{1}{N}\right)+\frac{\pi}{2}$$

$h(n)$奇对称 N 为偶数时，约束条件为

$$H_k=H_{N-k}$$

$$\theta_k=-k\pi\left(1-\frac{1}{N}\right)+\frac{\pi}{2}$$

（2）设计误差与哪些量有关，如何减小设计误差

从时域角度看，如果待设计的频率响应为 $H_d(e^{j\omega})$，对应的单位脉冲响应为

$$h_d(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\omega})e^{j\omega n}d\omega$$

在频域等间隔采样 N 点，利用 IDFT 得到的 $h(n)$是 $h_d(n)$以 N 为周期的周期性延拓

$$h(n)=\sum_{r=-\infty}^{\infty}h_d(n+rN)R_N(n)$$

如果 $H_d(e^{j\omega})$有间断点，那么 $h_d(n)$应是无限长的，这样一来由于时域混叠，引起所设计的 $h(n)$和 $h_d(n)$有偏差。为此在频域的采样点数 N 应加大。N 越大设计出的滤波器越逼近待设计的滤波器 $H_d(e^{j\omega})$。

从频域角度看 $H(e^{j\omega})=\sum_{k=0}^{N-1}H(k)\Phi\left(\omega-\frac{2\pi}{N}k\right)$，表明在采样点上 $H(e^{j\omega})$和 $H(k)$相等，逼近误差为零。在采样点之间其误差和 $H_d(e^{j\omega})$特性的平滑程度有关，特性越平滑误差越小，采样点数越大误差越小。

结论：为改善滤波器的特性，可以对频响间断点附近插入一个或几个过渡采样点，适当增加采样点数。

（3）频率采样法

对 $H_d(e^{j\omega})$进行频率采样得到$H(k)$，为了方便将$H(k)$表示成为幅度响应$|H(k)|$及相位响应$\theta(k)$的形式，即

$$H(k)=|H(k)|e^{j\theta(k)} \qquad k=0,1,\cdots,N-1$$

频率采样有两种方法：

① 频率采样法 1　第一个采样点在$\omega=0$处

$$H(k)=H_d(k)=H_d(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k}$$

$$h(n)=\frac{1}{N}\sum_{k=0}^{N-1}H(k)e^{j\frac{2\pi}{N}kn}$$

$$H(z)=\frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-W_N^{-k}z^{-1}}$$

$$H(e^{j\omega})=\frac{1}{N}e^{-j\left(\frac{N-1}{2}\right)\omega}\sum_{k=0}^{N-1}H(k)e^{-j\frac{\pi k}{N}}\frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2}-\frac{\pi k}{N}\right)}$$

由于

$$H(k)=\sum_{n=0}^{N-1}h(n)e^{-j\frac{2\pi}{N}kn}$$

当 $h(n)$为实数时，满足 $H(k)=H^*((N-k))_N$，故

$$|H(k)|=|H(N-k)|$$
$$\theta(k)=-\theta(N-k)$$

当N为奇数时，有线性相位约束条件

$$\theta(k)=\begin{cases}-\frac{2\pi}{N}k\left(\frac{N-1}{2}\right), & k=0,1,\cdots,\frac{N-1}{2}\\ \frac{2\pi}{N}(N-k)\left(\frac{N-1}{2}\right), & k=\frac{N+1}{2},\cdots,N-1\end{cases} \qquad \text{(1 型)}$$

$$\theta(k)=\begin{cases}\frac{\pi}{2}-\frac{2\pi}{N}k\left(\frac{N-1}{2}\right), & k=0,1,\cdots,\frac{N-1}{2}\\ -\frac{\pi}{2}+\frac{2\pi}{N}(N-k)\left(\frac{N-1}{2}\right), & k=\frac{N+1}{2},\cdots,N-1\end{cases} \qquad \text{(3 型)}$$

当 N 为偶数时，有

$$\theta(k)=\begin{cases}-\dfrac{2\pi}{N}k\left(\dfrac{N-1}{2}\right), & k=0,1,\cdots,\dfrac{N}{2}-1\\ \dfrac{2\pi}{N}(N-k)\left(\dfrac{N-1}{2}\right), & k=\dfrac{N}{2}+1,\cdots,N-1\\ 0, & k=\dfrac{N}{2}\end{cases} \qquad (2\text{ 型})$$

$$\theta(k)=\begin{cases}\dfrac{\pi}{2}-\dfrac{2\pi}{N}k\left(\dfrac{N-1}{2}\right), & k=0,1,\cdots,\dfrac{N}{2}-1\\ -\dfrac{\pi}{2}+\dfrac{2\pi}{N}(N-k)\left(\dfrac{N-1}{2}\right), & k=\dfrac{N}{2}+1,\cdots,N-1\\ 0, & k=\dfrac{N}{2}\end{cases} \qquad (4\text{ 型})$$

[例 11]　用频率采样法 1，设计 1 型 FIR 低通滤波器，要求的技术指标为

$$\omega_p=0.3\pi, A_p=5\text{dB}$$
$$\omega_r=0.4\pi, A_r=40\text{dB}$$

解：选择 N=21，则 ω_p 在 k=3 附近，ω_r 在 k=4 附近。

$|H(k)|$= {1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1}

$$\theta(k)=\begin{cases}-\dfrac{20\pi}{21}k, k=0,1,\cdots,10\\ \dfrac{20\pi}{21}(21-k), k=12,13,\cdots,20\end{cases}$$

$$H(k)=|H(k)|e^{j\theta(k)}$$

$h(n)$= IDFT[$H(k)$]

={−0.0414, 0, 0.0443, 0.0476, 0, −0.0606, −0.0732, 0, 0.1399, 0.2767, 0.3333, 0.2767, 0.1399, 0, −0.0732, −0.0606, 0, 0.0476, 0.0443, 0, −0.0414}

MATLAB 实现如下。

```
%s161.m
N=21;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;
Hrs=[ones(1,4),zeros(1,14),ones(1,3)];%理想振幅响应采样
Hdr=[1,1,0,0,1,1];wdl=[0,0.35,0.35,1.65,1.65,2];%理想振幅响应
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;%k取整数
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];%相位约束条件
H=Hrs.*exp(j*angH);%构成H(k)
h=real(ifft(H,N));%实际单位脉冲响应
[db,mag,pha,w]=freqz_m(h,[1]);
[Hr,ww,a,L]=hr_type1(h);%实际振幅响应
Ar=-round(max(db(200:1:501)))
subplot(221);plot(wl/pi,Hrs,'.',wdl,Hdr);title('频率样本H(k):N=21');
subplot(222);stem(l,h);title('实际单位脉冲响应h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi,Hrs,'.');title('实际振幅响应H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');
```

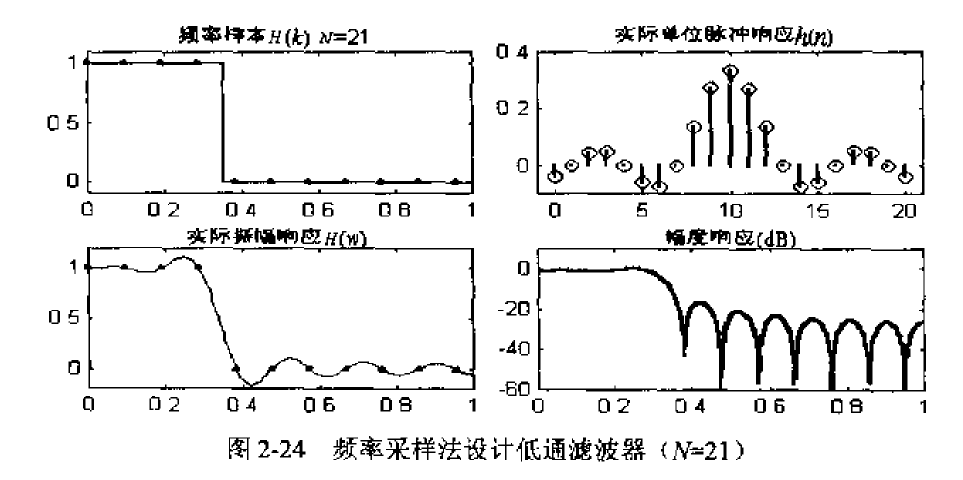


图 2-24 频率采样法设计低通滤波器（N=21）

从幅度响应曲线（图 2-24）可见，N=21 时，最小的阻带衰减为 14dB，没有达到设计要求。增大采样点数，N=61 时，最小的阻带衰减为 18dB，仍不满足指标，如图 2-25 所示。

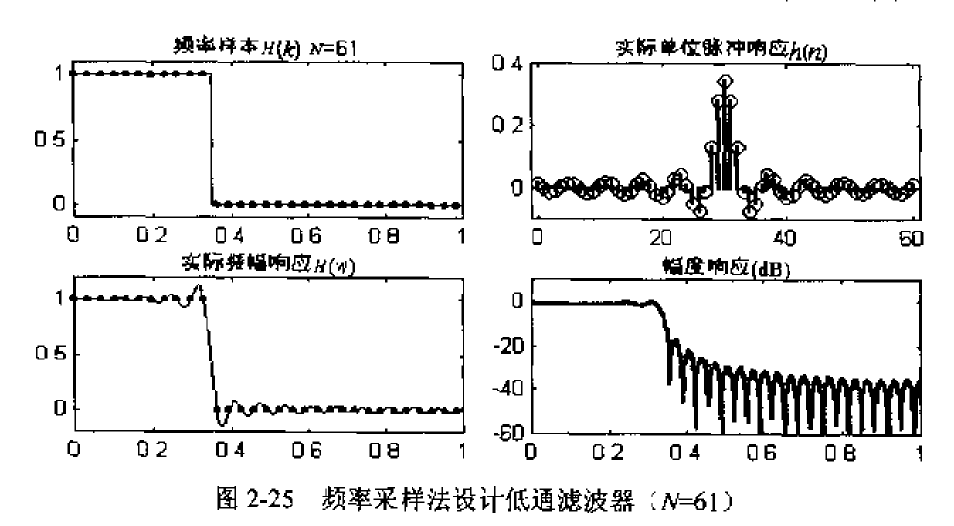


图 2-25 频率采样法设计低通滤波器（N=61）

在过渡带上加两个过渡采样点 T1 和 T2。

MATLAB 实现如下。

```
%s162.m
N=61;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;T1=0.7;T2=0.2;
Hrs=[ones(1,10),T1,T2,zeros(1,38),T2,T1,ones(1,9)];%理想振幅响应采样
Hdr=[1,1,0,0,1,1];wdl=[0,0.35,0.35,1.65,1.65,2];%理想振幅响应
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];%相位约束条件
H=Hrs.*exp(j*angH);%构成H(k)
h=real(ifft(H,N));%实际单位脉冲响应
[db,mag,pha,w]=freqz_m(h,[1]);
[Hr,ww,a,L]=hr_type1(h);%实际振幅响应
Ar=-round(max(db(200:1:501)))
subplot(221);plot(wl/pi,Hrs,'.',wdl,Hdr);title('频率样本H(k):N=21')
subplot(222);stem(l,h);title('实际单位脉冲响应h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi,Hrs,'.');title('实际振幅响应H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');
```

结果见图 2-26。

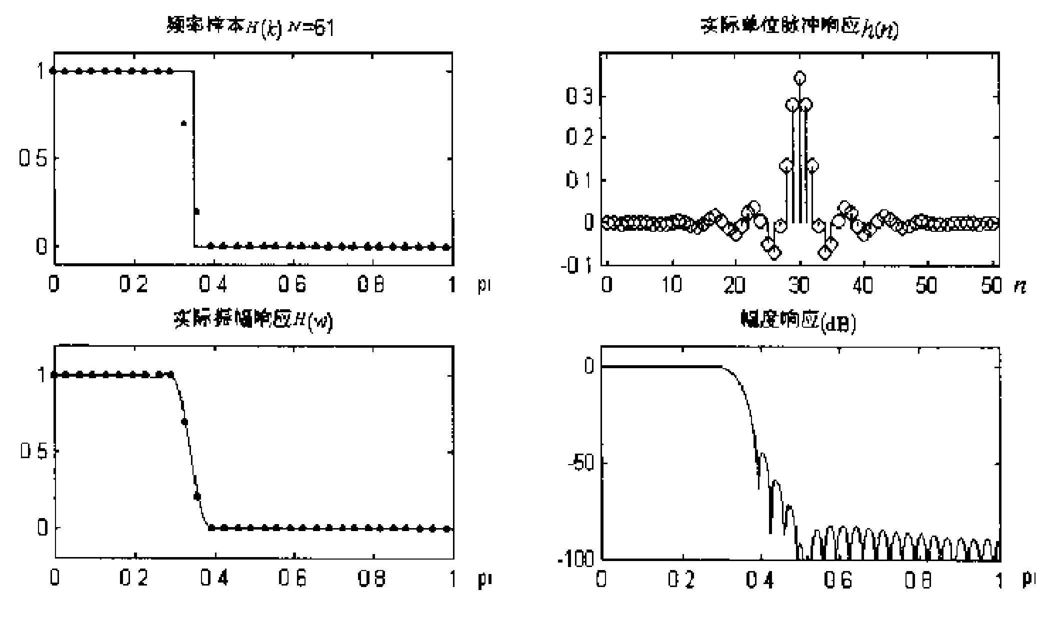


图 2-26 加过渡点的频率采样法设计低通滤波器(1 型)（N=61）

从幅度响应曲线可见，N=61 时，最小的阻带衰减为 45dB，达到设计要求。

[例 12] 用频率采样法 1，设计 2 型 FIR 低通滤波器，要求的技术指标为

$$\omega_p = 0.3\pi, A_p = 1\text{dB}$$

$$\omega_r = 0.4\pi, A_r = 40\text{dB}$$

解：选择 N=60，则ω_p在 k=9 处，ω_r在 k=12 处。$T1$=0.7，$T2$=0.2。

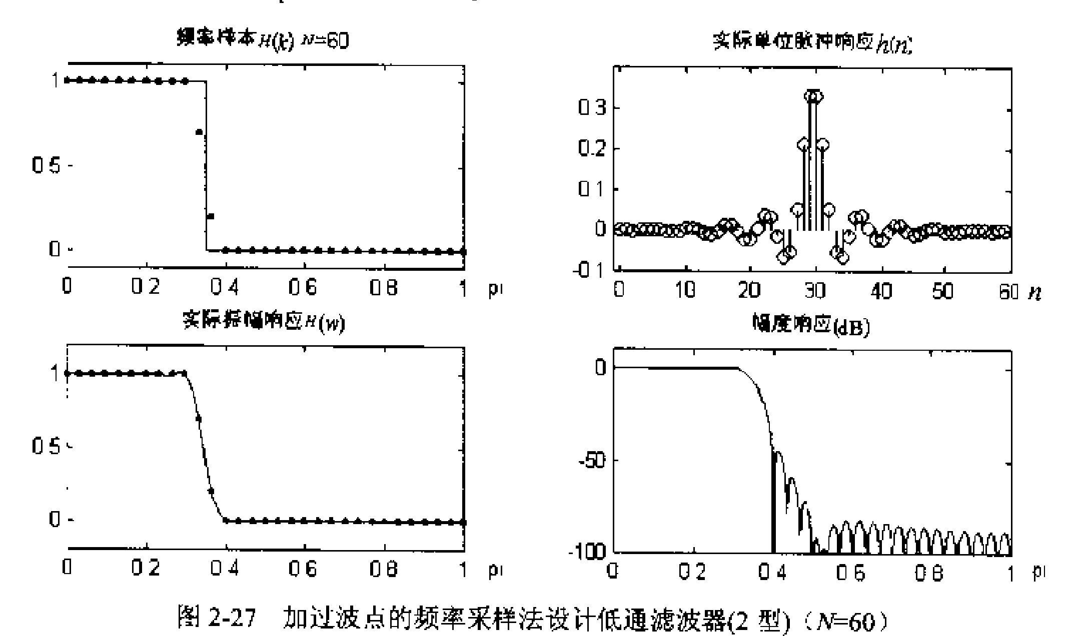


图 2-27 加过波点的频率采样法设计低通滤波器(2 型)（N=60）

```
%s163.m
N=60;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;T1=0.7;T2=0.2;
Hrs=[ones(1,10),T1,T2,zeros(1,37),T2,T1,ones(1,9)];%理想振幅响应采样
Hdr=[1,1,0,0,1,1];wdl=[0,0.35,0.35,1.65,1.65,2];%理想振幅响应
k1=0:floor((N-1)/2);k2=floor((N-1)/2+1):N-1;%k 取整数
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];%相位约束条件
H=Hrs.*exp(j*angH);%构成 H(k)
```

```
h=real(ifft(H,N));%实际单位脉冲响应
[db,mag,pha,w]=freqz_m(h,[1]);
[Hr,ww,a,L]=hr_type2(h);%实际振幅响应
Ap=-db(150): Ar=-round(max(db(200:1:501)))
subplot(221);plot(wl/pi,Hrs,'.',wdl,Hdr);title('频率样本H(k):N=60')
subplot(222);stem(l,h);title('实际单位脉冲响应h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi,Hrs,'.');title('实际振幅响应H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');
```

结果见图 2-27。

从幅度响应曲线可见，N=60 时，最小的阻带衰减为 45dB，达到设计要求。

[例 13]　用频率采样法 1，设计 1 型 FIR 高通滤波器，要求的技术指标为

$$\omega_p = 0.8\pi, A_p = 1\text{dB}$$

$$\omega_r = 0.7\pi, A_r = 40\text{dB}$$

解：选择 N=61，则ω_p在 k=24 处，ω_r在 k=21 处。

选择 $T1$=0.1095，$T2$=0.598。

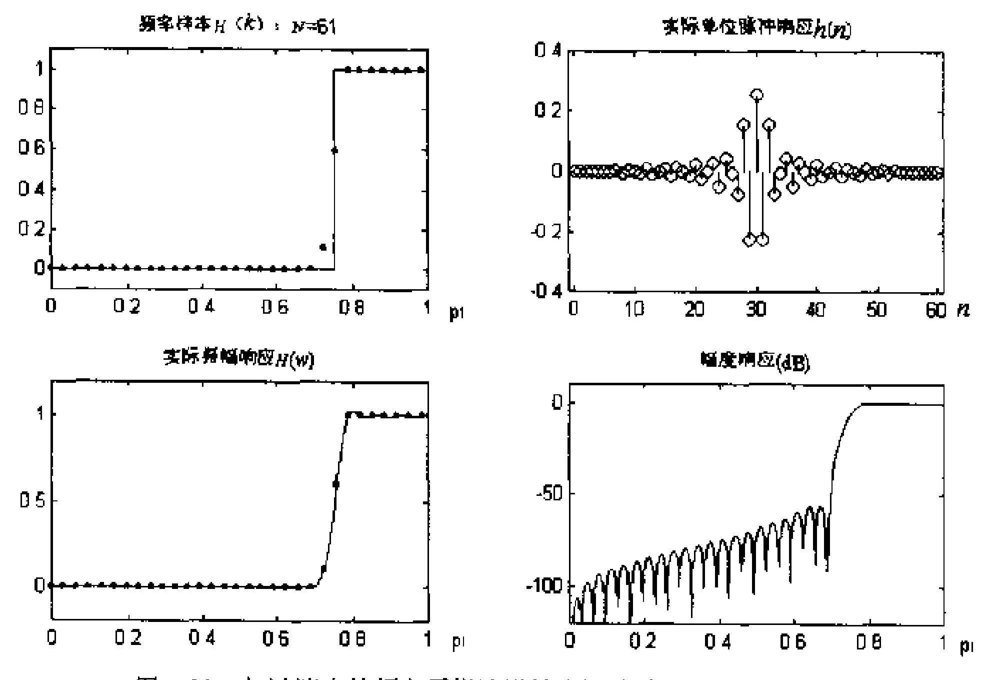


图 2-28　加过渡点的频率采样法设计高通滤波器(1 型)（N=61）

```
%s164.m
N=61;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;
T1=0.1095;T2=0.598;
Hrs=[zeros(1,22),T1,T2,ones(1,14),T2,T1,zeros(1,21)];
Hdr=[0,0,1,1];wdl=[0,0.75,0.75,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];
H=Hrs.*exp(j*angH);
h=real(ifft(H,N));
[db,mag,pha,w]=freqz_m(h,1);
[Hr,ww,a,L]=hr_type1(h);
```

```
subplot(221);plot(wl/pi,Hrs,'.',wdl,Hdr);title('频率样本 H (k): N=61')
subplot(222);stem(l,h);title('实际单位脉冲响应 h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi,Hrs,'.');title('实际振幅响应 H(w)');
subplot(224);plot(w/pi,db);title('幅度响应 (dB)');
```

结果见图 2-28。

从幅度响应曲线可见，$N=61$ 时，最小的阻带衰减为 50dB，达到设计要求。

[例 14] 用频率采样法 1，设计 4 型 FIR 高通滤波器，要求的技术指标为

$$\omega_p = 0.8\pi, A_p = 1\text{dB}$$

$$\omega_r = 0.7\pi, A_r = 40\text{dB}$$

解：选择 $N=60$，则 ω_p 在 $k=24$ 处，ω_r 在 $k=21$ 处。

选择 $T1=0.1095$，$T2=0.598$。

```
%s164.m
N=60;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;
T1=0.1095;T2=0.598;
Hrs=[zeros(1,22),T1,T2,ones(1,13),T2,T1,zeros(1,21)];
Hdr=[0,0,1,1];wdl=[0,0.75,0.75,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[pi/2-alpha*(2*pi)/N*k1,-pi/2+alpha*(2*pi)/N*(N-k2)];
H=Hrs.*exp(j*angH);
h=real(ifft(H,N));
[db,mag,pha,w]=freqz_m(h,1);
[Hr,ww,d,L]=hr_type4(h);
subplot(221);plot(wl/pi,Hrs,'.',wdl,Hdr);title('频率样本 H (k): N=60')
subplot(222);stem(l,h);title('实际单位脉冲响应 h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi,Hrs,'.');title('实际振幅响应 H(w)');
subplot(224);plot(w/pi,db);title('幅度响应 (dB)');
```

结果见图 2-29。

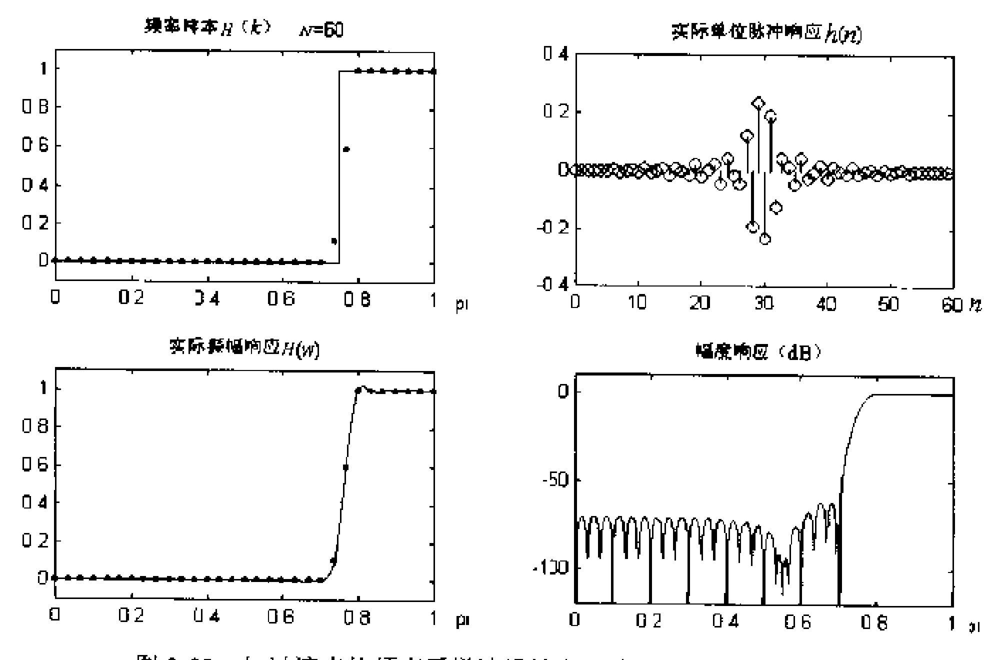


图 2-29 加过渡点的频率采样法设计高通滤波器(4 型)（$N=60$）

从幅度响应曲线可见，$N=60$ 时，最小的阻带衰减为 50dB，达到设计要求。

[例 15] 用频率采样法 1，设计 2 型 FIR 带通滤波器，要求的技术指标为

$$\omega_{r1}=0.2\pi, A_{r1}=20\text{dB}$$
$$\omega_{p1}=0.3\pi, A_{p1}=2\text{dB}$$
$$\omega_{p2}=0.6\pi, A_{p2}=2\text{dB}$$
$$\omega_{r2}=0.7\pi, A_{r2}=20\text{dB}$$

解：选择 $N=40$，则ω_{p1}在 $k=6$ 处，ω_{r1}在 $k=4$ 处，ω_{p2}在 $k=12$ 处，ω_{r2}在 $k=14$ 处。选择一个过渡样点 $T1=0.1095$。

```
%s166.m
N=40;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;
T1=0.1095;
Hrs=[zeros(1,5),T1,ones(1,7),T1,zeros(1,13),T1,ones(1,7),T1,zeros(1,4)];
Hdr=[0,0,1,1,0,0];wdl=[0,0.25,0.25,0.65,0.65,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];
H=Hrs.*exp(j*angH);
h=real(ifft(H,N));
[db,mag,pha,w]=freqz_m(h,1);
[Hr,ww,b,L]=hr_type2(h);
subplot(221);plot(wl/pi,Hrs,'.',wdl,Hdr); title('频率样木H(k): N=40')
subplot(222);stem(l,h);title('实际单位脉冲响应h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi,Hrs,'.');title('实际振幅响应H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');
```

结果见图 2-30。

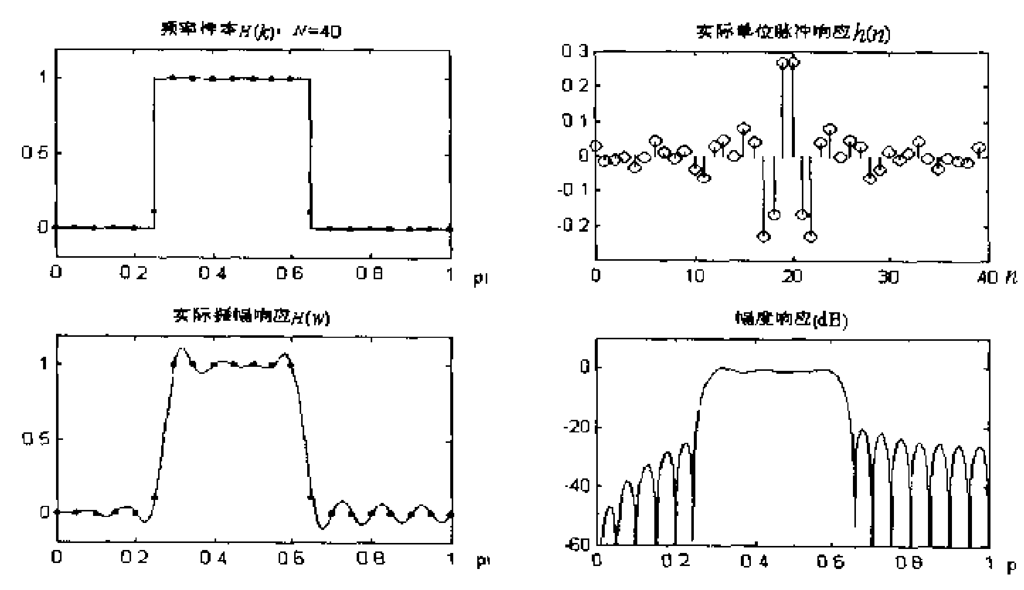


图 2-30 加过渡点的频率采样法设计带通滤波器(2 型)（$N=40$）

从幅度响应曲线可见，$N=40$ 时，最小的阻带衰减为 22dB，达到设计要求。

[例 16] 用频率采样法 1，设计 1 型 FIR 带阻滤波器，要求的技术指标为

$$\omega_{p1}=0.2\pi, A_{p1}=2\text{dB}$$

$$\omega_{r1} = 0.3\pi, A_{r1} = 20\text{dB}$$
$$\omega_{r2} = 0.6\pi, A_{r2} = 20\text{dB}$$
$$\omega_{p2} = 0.7\pi, A_{p2} = 2\text{dB}$$

解：选择N=41，则ω_{p1}在k=4处，ω_{r1}在k=6处，ω_{p2}在k=14处，ω_{r2}在k=12处。

选择一个过渡样点$T1$=0.1095。

```
%s167.m
N=41;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;
T1=0.1095;
Hrs=[ones(1,5),T1,zeros(1,7),T1,ones(1,14),T1,zeros(1,7),T1,ones(1,4)];
Hdr=[1,1,0,0,1,1];wdl=[0,0.25,0.25,0.65,0.65,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];
H=Hrs.*exp(j*angH);
h=real(ifft(H,N));
[db,mag,pha,w]=freqz_m(h,1);
[Hr,ww,a,L]=hr_type1(h);
Ap=-min(db(350:501)); Ar=-round(max(db(130:300)))
subplot(221);plot(wl/pi,Hrs,'.',wdl,Hdr); title('频率样本H(k): N=41')
subplot(222);stem(l,h);title('实际单位脉冲响应h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi,Hrs,'.');title('实际振幅响应H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');
```

结果见图2-31。

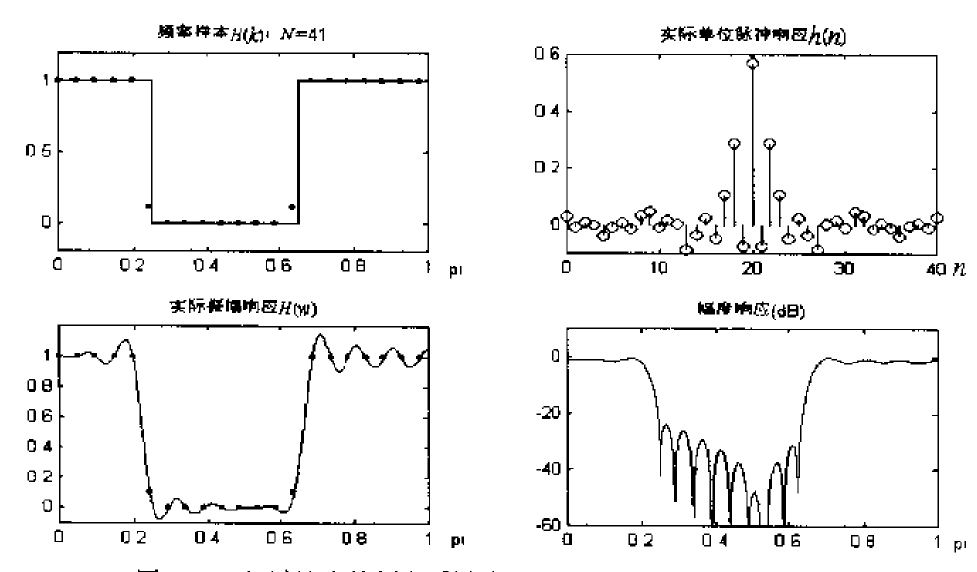


图2-31 加过渡点的频率采样法设计带阻滤波器(1型)（N=41）

从幅度响应曲线可见，N=41时，最小的阻带衰减为24dB，达到设计要求。

② 频率采样法2 第一个采样点在$\omega = \dfrac{\pi}{N}$处。

$$H(k) = H_d(k) = H_d(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k+\frac{\pi}{N}}$$

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k) e^{j\left(\frac{2\pi}{N}kn+\frac{\pi}{N}\right)} = e^{j\frac{\pi}{N}} \text{IDFT}[H(k)]$$

$$H(z)=\frac{1+z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-W_N^{-\left(k+\frac{1}{2}\right)}z^{-1}}$$

$$H(e^{j\omega})=\frac{\cos\left(\frac{\omega N}{2}\right)}{N}e^{-j\left(\frac{N-1}{2}\right)\omega}\sum_{k=0}^{N-1}\frac{H(k)e^{-j\frac{\pi}{N}\left(k+\frac{1}{2}\right)}}{j\sin\left[\frac{\omega}{2}-\frac{\pi}{N}\left(k+\frac{1}{2}\right)\right]}$$

由于

$$H(k)=\sum_{n=0}^{N-1}h(n)e^{-j\frac{2\pi}{N}\left(k+\frac{1}{2}\right)n}$$

当 $h(n)$为实数时，满足$H(k)=H^*((N-1-k))_N$，故

$$|H(k)|=|H(N-1-k)|$$
$$\theta(k)=-\theta(N-1-k)$$

当 N 为奇数时，有线性相位约束条件

$$\theta(k)=\begin{cases}-\frac{2\pi}{N}\left(k+\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=0,1,\cdots,\frac{N-3}{2}\\ 0, & k=\frac{N-1}{2}\\ \frac{2\pi}{N}\left(N-k-\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=\frac{N+1}{2},\cdots,N-1\end{cases} \quad (1\text{型})$$

$$\theta(k)=\begin{cases}\frac{\pi}{2}-\frac{2\pi}{N}\left(k+\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=0,1,\cdots,\frac{N-3}{2}\\ 0, & k=\frac{N-1}{2}\\ -\frac{\pi}{2}+\frac{2\pi}{N}\left(N-k-\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=\frac{N+1}{2},\cdots,N-1\end{cases} \quad (3\text{型})$$

当 N 为偶数时，有

$$\theta(k)=\begin{cases}-\frac{2\pi}{N}\left(k+\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=0,1,\cdots,\frac{N}{2}-1\\ \frac{2\pi}{N}\left(N-k-\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=\frac{N}{2},\cdots,N-1\end{cases} \quad (2\text{型})$$

$$\theta(k)=\begin{cases}\frac{\pi}{2}-\frac{2\pi}{N}\left(k+\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=0,1,\cdots,\frac{N}{2}-1\\ -\frac{\pi}{2}+\frac{2\pi}{N}\left(N-k-\frac{1}{2}\right)\left(\frac{N-1}{2}\right), & k=\frac{N}{2},\cdots,N-1\end{cases} \quad (4\text{型})$$

[例 17] 用频率采样法 2，设计 I 型 FIR 低通滤波器，要求的技术指标为

$$\omega_p=0.25\pi, A_p=2\text{dB}$$
$$\omega_r=0.35\pi, A_r=20\text{dB}$$

解：选择 N=41，则ω_p在 k=5 附近，ω_r在 k=7 附近。

```
%s168.m
N=41;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;
Hrs=[ones(1,6),0.1,zeros(1,27),0.1,ones(1,6)];
```

```
Hdr=[1,1,0,0];wdl=[0,0.3,0.3,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[-alpha*(2*pi)/N*(k1+0.5),alpha*(2*pi)/N*(N-k2-0.5)];
H=Hrs.*exp(j*angH);
h1=ifft(H,N);n=0:1:N-1;
h=real(h1.*exp(j*pi*n/N));
[db,mag,pha,w]=freqz_m(h,[1]);
[Hr,ww,a,L]=hr_type1(h);
subplot(221);plot(wl/pi+1/N,Hrs,'.',wdl,Hdr);title('频率样本 H(k): N=41');
subplot(222);stem(l,h);title('实际单位脉冲响应 h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi+1/N,Hrs,'.');title('实际振幅响应 H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');axis([0 1 -80 10]);
```

结果见图 2-32。

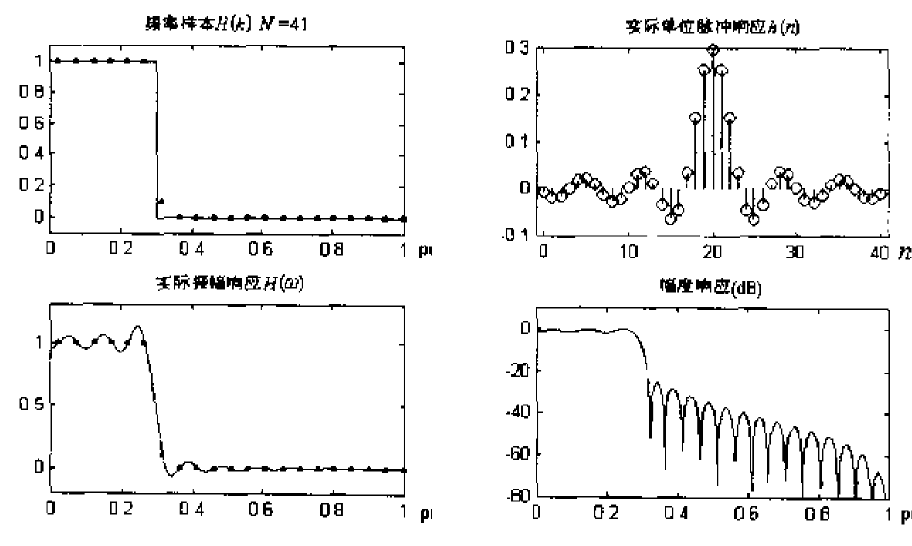


图 2-32 加过渡点的频率采样法设计低通滤波器(1 型)（N=41）

从幅度响应曲线可见，N=41 时，最小的阻带衰减为 24dB，达到设计要求。

[例 18] 用频率采样法 2，设计 1 型 FIR 高通滤波器，要求的技术指标为

$$\omega_p = 0.35\pi, A_p = 2\text{dB}$$

$$\omega_r = 0.25\pi, A_r = 20\text{dB}$$

解：选择 N=41，则 ω_p 在 k=7 附近，ω_r 在 k=5 附近。选择一个过渡点 $T1$=0.1；

```
%s169.m
N=41;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l; T1=0.1;
Hrs=[zeros(1,6), T1,ones(1,27), T1,zeros(1,6)];
Hdr=[0,0,1,1];wdl=[0,0.3,0.3,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[-alpha*(2*pi)/N*(k1+0.5),alpha*(2*pi)/N*(N-k2-0.5)];
H=Hrs.*exp(j*angH);
h1=ifft(H,N);
n=0:1:N-1;
h=real(h1.*exp(j*pi*n/N));
[db,mag,pha,w]=freqz_m(h,[1]);
[Hr,ww,a,L]=hr_type1(h);
```

```
subplot(221);plot(wl/pi+1/N,Hrs,'.',wdl,Hdr);title('频率样本 H(k): N=41');
subplot(222);stem(l,h);title('实际单位脉冲响应 h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi+1/N,Hrs,'.');title('实际振幅响应 H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');axis([0 1 -80 10]);
```

结果见图 2-33。

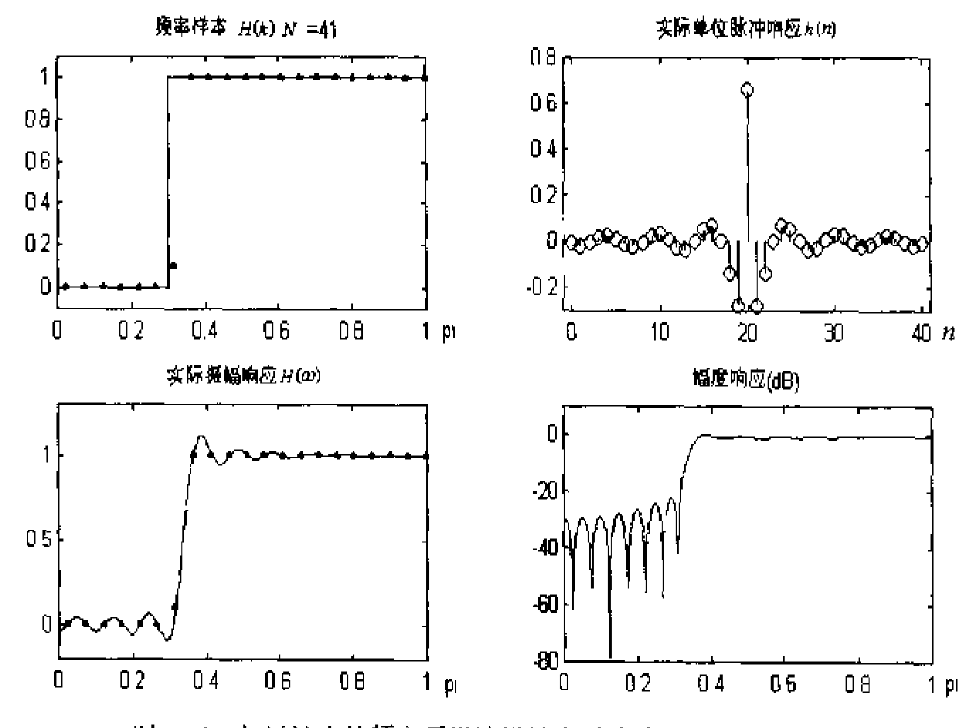


图 2-33 加过渡点的频率采样法设计高通滤波器(1 型)（N=41）

从幅度响应曲线可见，N=41 时，最小的阻带衰减为 22dB，达到设计要求。

[例 19] 用频率采样法 2，设计 4 型 FIR 高通滤波器，要求的技术指标为

$$\omega_p = 0.35\pi, A_p = 2\text{dB}$$

$$\omega_r = 0.25\pi, A_r = 20\text{dB}$$

解：选择 N=40，则ω_p在 k=7 附近，ω_r在 k=5 附近。

```
%s170.m
N=40;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;
Hrs=[zeros(1,6),0.1,ones(1,26),0.1,zeros(1,6)];
Hdr=[0,0,1,1];wdl=[0,0.3,0.3,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[pi/2-alpha*(2*pi)/N*(k1+0.5),-pi/2+alpha*(2*pi)/N*(N-k2-0.5)];
H=Hrs.*exp(j*angH);
h1=ifft(H,N);
n=0:1:N-1;
h=real(h1.*exp(j*pi*n/N));
[db,mag,pha,w]=freqz_m(h,[1]);
[Hr,ww,d,L]=hr_type4(h);
subplot(221);plot(wl/pi+1/N,Hrs,'.',wdl,Hdr);title('频率样本 H(k): N=40');
subplot(222);stem(l,h);title('实际单位脉冲响应 h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi+1/N,Hrs,'.');title('实际振幅响应 H(w)');
```

```
subplot(224);plot(w/pi,db);title('幅度响应(dB)');axis([0 1 -80 10]);
```

结果见图 2-34。

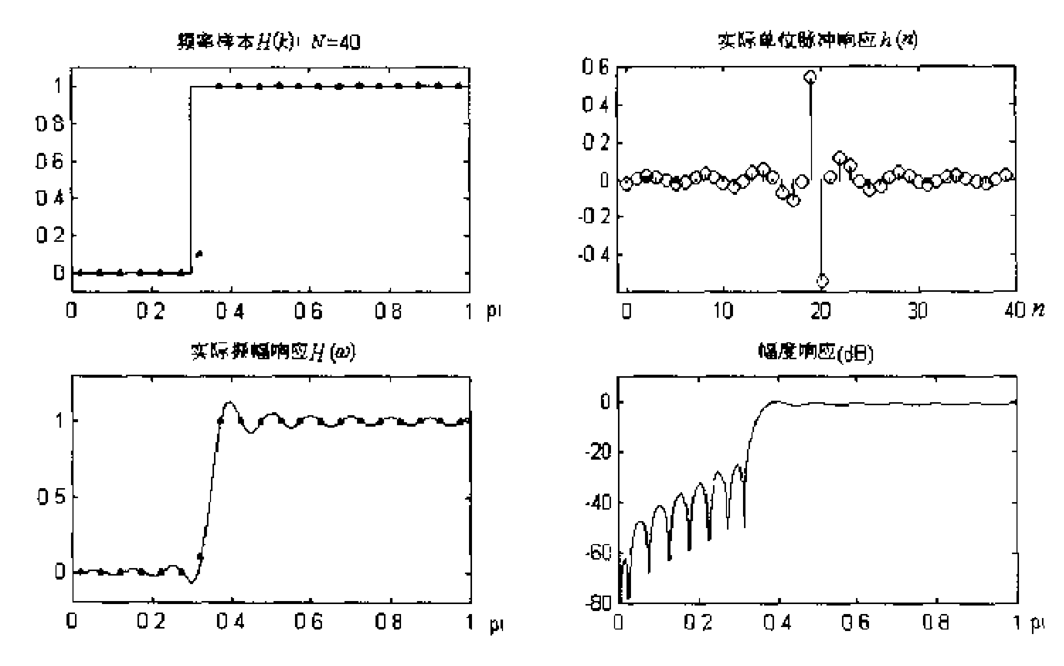


图 2-34　加过渡点的频率采样法设计高通滤波器(4 型)（N=40）

从幅度响应曲线可见，N=40 时，最小的阻带衰减为 25dB，达到设计要求。

[例 20]　用频率采样法 2，设计 3 型 FIR 带通滤波器，要求的技术指标为

$$\omega_{r1} = 0.25\pi, A_{r1} = 20\text{dB}, \omega_{p1} = 0.35\pi, A_{p1} = 2\text{dB}$$

$$\omega_{p2} = 0.65\pi, A_{p2} = 2\text{dB}, \omega_{r2} = 0.75\pi, A_{r2} = 20\text{dB}$$

解：选择 N=41，则ω_{p1}在 k=7 处，ω_{r1}在 k=5 处，ω_{p1}在 k=13 处，ω_{r2}在 k=15 处。

选择一个过渡样点 $T1$=0.1095。

```
%s171.m
N=41;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;T1=0.1;
Hrs=[zeros(1,6),T1,ones(1,7),T1,zeros(1,11),T1,ones(1,7),T1,zeros(1,6)];
Hdr=[0,0,1,1,0,0];wdl=[0,0.3,0.3,0.7,0.7,1];
k1=0:floor((N-1)/2);k2=floor((N-1)/2)+1:N-1;
angH=[pi/2-alpha*(2*pi)/N*(k1+0.5),-pi/2+alpha*(2*pi)/N*(N-k2-0.5)];
H=Hrs.*exp(j*angH);
h1=ifft(H,N);
n=0:1:N-1;
h=real(h1.*exp(j*pi*n/N));
[db,mag,pha,w]=freqz_m(h,[1]);
[Hr,ww,c,L]=hr_type3(h);
subplot(221);plot(wl/pi+1/N,Hrs,'.',wdl,Hdr);title('频率样本H(k)：N=41');
subplot(222);stem(l,h);title('实际单位脉冲响应h(n)');
subplot(223);plot(ww/pi,Hr,wl/pi+1/N,Hrs,'.');title('实际振幅响应H(w)');
subplot(224);plot(w/pi,db);title('幅度响应(dB)');axis([0 1 -80 10]);
```

结果见图 2-35。

从幅度响应曲线可见，N=41 时，最小的阻带衰减为 22dB，达到设计要求。

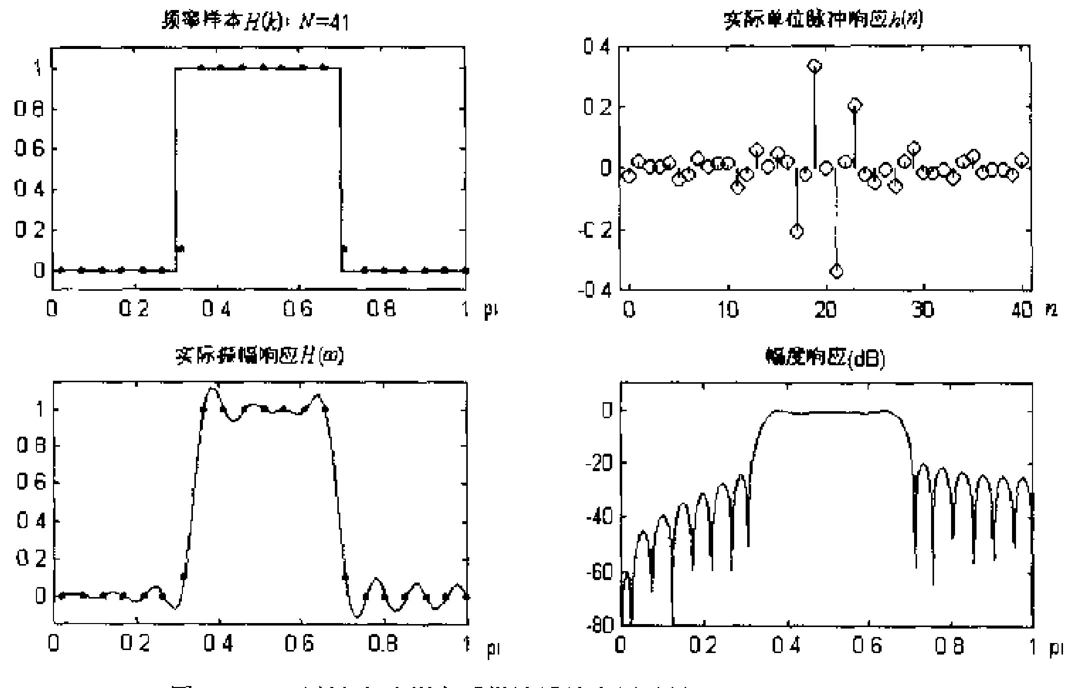


图 2-35 加过渡点的频率采样法设计高通滤波器(3 型) (N=41)

2.2.4 FIR 滤波器的应用实例

[例 21] 用频率采样法 1 设计具有线性相位，技术指标为

$$\omega_p = 0.3\pi, A_p = 1\text{dB}$$

$$\omega_r = 0.4\pi, A_r = 40\text{dB}$$

2 型 FIR 低通滤波器，并对方波信号进行滤波。

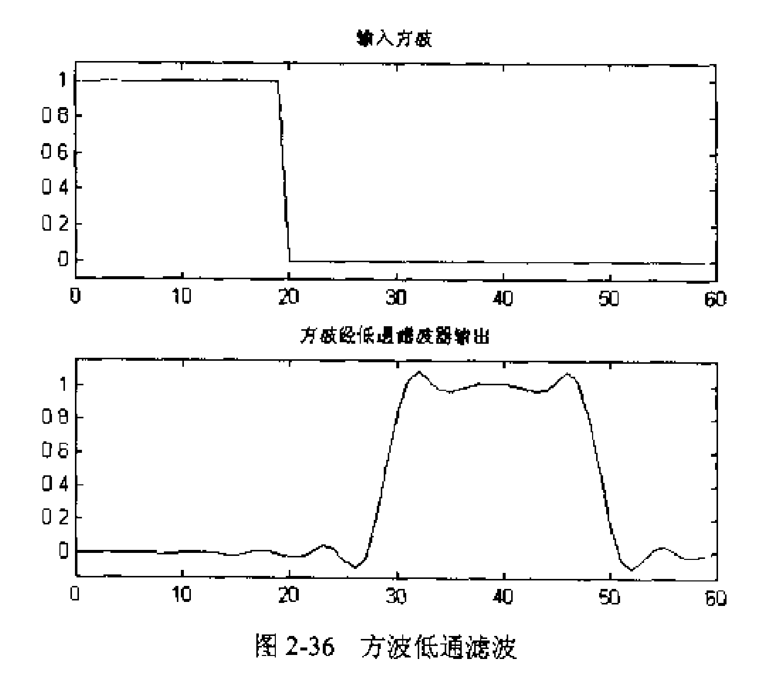


图 2-36 方波低通滤波

```
%s172.m (s163.m 设计的低通滤波器)
N=60;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;T1=0.7;T2=0.2;
Hrs=[ones(1,10),T1,T2,zeros(1,37),T2,T1,ones(1,9)];%理想振幅响应采样
k1=0:floor((N-1)/2);k2=floor((N-1)/2+1):N-1;%k 取整数
```

```
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];%相位约束条件
H=Hrs.*exp(j*angH);%构成H(k)
h=real(ifft(H,N));%实际单位脉冲响应
x=[ones(1,20),zeros(1,N-20)];
t=0:length(x)-1;
subplot(211);plot(t,x);title('输入方波');
[b0,B,A]=cir2cas(h,1);
y=casfiltr(b0,B,A,x);%级联形式实现
subplot(212);plot(t,y);title('方波经低通滤波器输出');
```

结果见图 2-36。

从图 2-36 可见，输出信号相对输入信号有 α=29.5 的延时，保留了低频成分，去掉了高频成分。达到数字滤波的目的。

[例 22] 用窗函数法设计具有线性相位，技术指标为

$$\omega_p = 0.2\pi, A_p = 0.25\text{dB}$$
$$\omega_r = 0.4\pi, A_r = 50\text{dB}$$

2 型 FIR 低通滤波器，并对方波信号进行滤波。

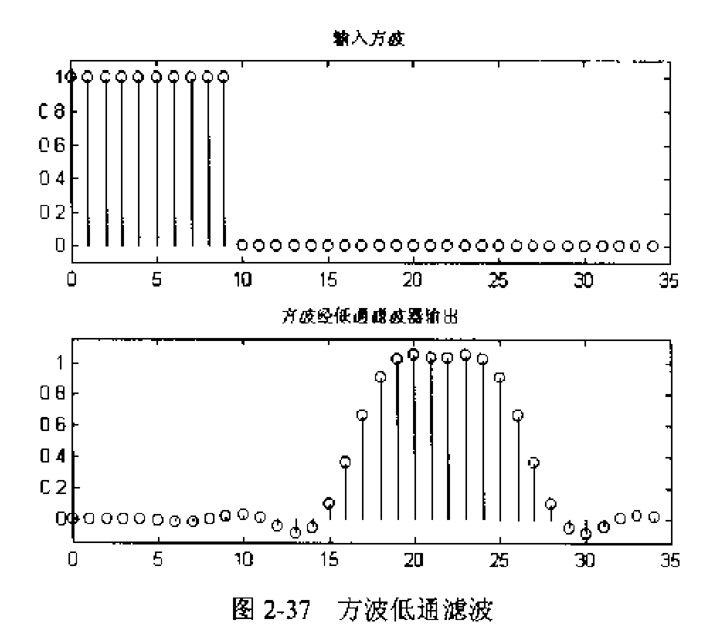


图 2-37 方波低通滤波

```
%s173.m (s155.m 设计的低通滤波器)
wp=0.2*pi;wr=0.4*pi;
tr_width=wr-wp;
N=ceil(6.6*pi/tr_width)+2
n=0:1:N-1;
wc=(wr+wp)/2;
hd=ideal_lp(wc,N);
w_ham=(hamming(N))';
h=hd.*w_ham;
x=[ones(1,10),zeros(1,N-10)];
t=0:length(x)-1;
subplot(211);stem(t,x);title('输入方波');
```

```
[b0,B,A]=dir2cas(h,1);
y=casfiltr(b0,B,A,x);%级联形式实现
subplot(212);stem(t,y);title('方波经低通滤波器输出');
```

从图 2-37 中可见，输出信号相对输入信号有α=17 的延时，保留了低频成分，去掉了高频成分，达到数字滤波的目的。

[例 23] 一个含有高频噪声的信号保存在数据文件 noisbloc 中，试用频率采样法 1 设计的低通滤波器对其进行除噪。

```
%s174.m
N=100;alpha=(N-1)/2;l=0:N-1;wl=(2*pi/N)*l;T1=0.7;T2=0.2;
Hrs=[ones(1,20),T1,T2,zeros(1,57),T2,T1,ones(1,19)];
k1=0:floor((N-1)/2);k2=floor((N-1)/2+1):N-1;
angH=[-alpha*(2*pi)/N*k1,alpha*(2*pi)/N*(N-k2)];
H=Hrs.*exp(j*angH);
h=real(ifft(H,N));
load noisbloc;indx=1:1000;
x=noisbloc(indx);
subplot(211);plot(x);title('含噪声的输入信号');
y=filter(h,1,x);%直接型实现
subplot(212);plot(y);title('除噪后的输出信号');
```

结果如图 2-38，可见用滤波器进行除噪的方法可以对信号的噪声按要求进行抑制。

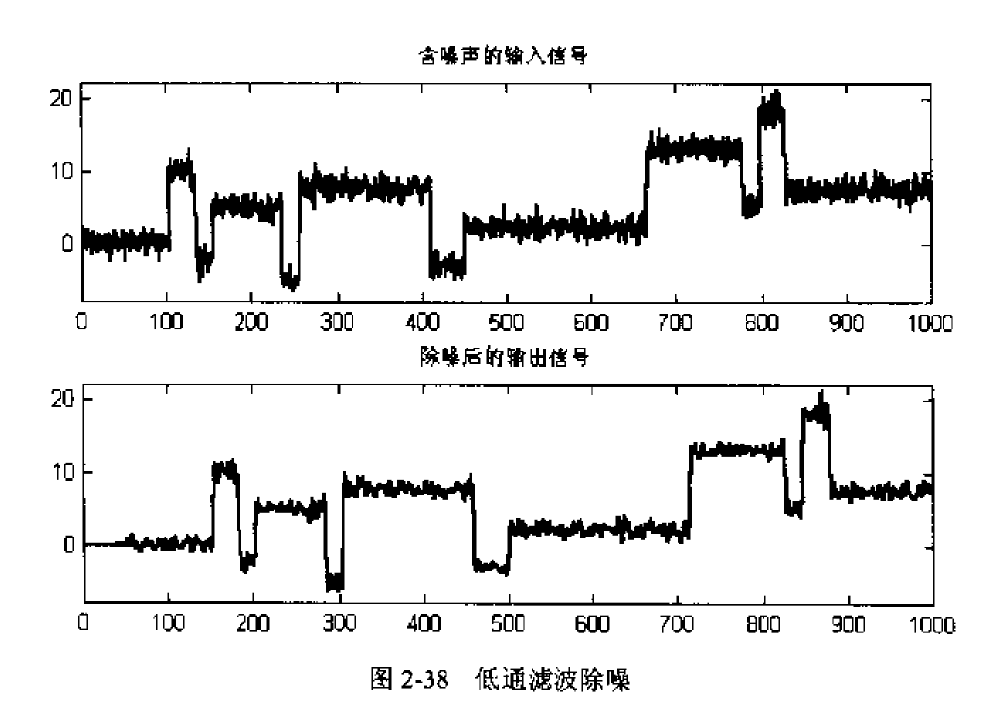


图 2-38 低通滤波除噪

2.3 IIR 滤波器设计

IIR 数字滤波器的单位脉冲响应是无限长的，而所有的模拟滤波器一般都具有无限长的单位脉冲响应，因此它与模拟滤波器相匹配。模拟滤波器的理论和设计方法已发展得相当成熟，有典型的模拟滤波器供选择，如巴特沃斯（Butterworth）模拟滤波器、切比雪夫（Chebyshev）模拟滤波器、椭圆（Cauer）模拟滤波器等。这些滤波器各有特点，根据其特

点用相应的公式，按规定的指标设计出模拟滤波器。IIR 数字滤波器设计的基本技术就是利用复值映射把已知的模拟滤波器转换成数字滤波器。

IIR 数字滤波器可用 N 阶差分方程

$$y(n)=\sum_{i=0}^{M}b_i x(n-i)-\sum_{k=1}^{N}a_k y(n-k)$$

来描述，相应的系统函数为

$$H(z)=\frac{\sum_{i=0}^{M}b_i z^{-i}}{1+\sum_{i=1}^{N}a_k z^{-k}}$$

滤波器的设计任务就是根据预先确定的技术指标，找到一组系数 a_k、b_i，使滤波器的性能满足其技术指标。

2.3.1 模拟滤波器原型的特征

2.3.1.1 模拟滤波器的技术指标及逼近方法

模拟滤波器按频带分成低通滤波器、高通滤波器、带通滤波器和带阻滤波器。设模拟滤波器的系统函数为 $H_a(s)$，频率特性为 $H_a(j\Omega)$。则$|H_a(j\Omega)|$是滤波器的幅度特性，如图 2-39 所示。

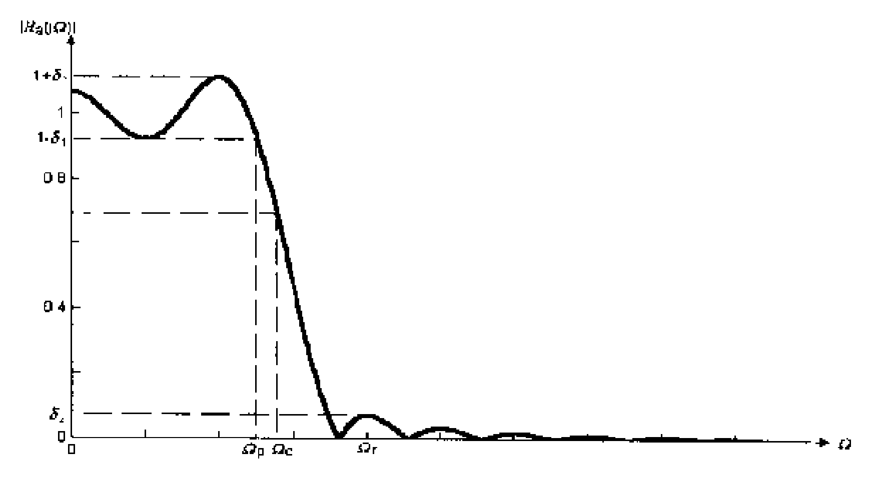


图 2-39 模拟低通滤波器技术指标

模拟滤波器的技术指标：

① 通带截止频率 Ω_p；

② 阻带起始频率 Ω_r；

③ 通带允许的最大衰减 A_p；

④ 阻带允许的最小衰减 A_r；

⑤ 3dB 截止频率 Ω_c。

模拟滤波器的技术指标给定以后，设计模拟滤波器的任务就是设计一个系统函数 $H_a(s)$，使其满足技术指标。一般用幅度平方函数作为逼近函数，因此根据幅度平方函数来确定 $H_a(s)$。

2.3.1.2 巴特沃斯模拟滤波器

巴特沃斯模拟滤波器的特性和系统函数

巴特沃斯模拟低通滤波器的幅度平方函数定义为

$$|H_a(j\Omega)|^2=\frac{1}{1+\left(\frac{j\Omega}{j\Omega_c}\right)^{2N}}$$

式中 N ——滤波器的阶次，为整数；

Ω_c——3dB 截止频率，rad/s。

巴特沃斯模拟滤波器的特性是：

① 在通带中有最大平坦的幅度特性；

② 在阻带内的逼近是单调变化的；

③ 当 N 越大，滤波器的特性越接近理想的矩形幅度特性。

将幅度平方函数写成 s 的函数，得

$$H_a(s)H_a(-s)=\frac{1}{1+\left(\frac{s}{j\Omega_c}\right)^{2N}}$$

极点有 $2N$ 个

$$p_k=(-1)^{\frac{1}{2N}}(j\Omega_c)=\Omega_c e^{j\frac{N+2k+1}{2N}\pi} \qquad k=0,1,2,\cdots,2N-1$$

取 s 左平面的 N 个极点组成 $H_a(s)$

$$H_a(s)=\frac{(-1)^N\Omega_c^{\ N}}{\prod\limits_k(s-p_k)}$$

巴特沃斯模拟滤波器的设计方法

设计的基本思想是在Ω_p，Ω_r处满足指标 A_p，A_r

$$A_p=10\lg\left[1+\left(\frac{\Omega_p}{\Omega_c}\right)^{2N}\right]$$

$$A_r=10\lg\left[1+\left(\frac{\Omega_r}{\Omega_c}\right)^{2N}\right]$$

解上面两个方程得到阶次 N 和截止频率Ω_c

$$N\geqslant\frac{\lg[(10^{A_p/10}-1)/(10^{A_r/10}-1)]}{2\lg(\Omega_p/\Omega_r)}$$

在Ω_p处精确地满足指标有

$$\Omega_c=\frac{\Omega_p}{\sqrt[2N]{(10^{A_p/10}-1)}}$$

在Ω_r处精确地满足指标有

$$\Omega_c=\frac{\Omega_r}{\sqrt[2N]{(10^{A_r/10}-1)}}$$

根据阶次 N 和截止频率Ω_c，可得到巴特沃斯模拟滤波器的系统函数 $H_a(s)$。

[例 1] 设计一个低通巴特沃斯模拟滤波器，要求

通带截止频率：$\Omega_p = 0.2\pi$，通带最大衰减：A_p=10dB

阻带起始频率：$\Omega_r = 0.3\pi$，阻带最小衰减：A_r=20dB

解：

$$N \geqslant \frac{\lg[(10^{A_p/10}-1)/(10^{A_r/10}-1)]}{2\lg(\Omega_p/\Omega_r)} = \frac{\lg[(10^{10/10}-1)/(10^{20/10}-1)]}{2\lg\left(\dfrac{0.2\pi}{0.3\pi}\right)} = 2.957$$

$$\Omega_c = \frac{\Omega_p}{\sqrt[2N]{(10^{A_p/10}-1)}} = \frac{0.2\pi}{\sqrt[6]{(10^{10/10}-1)}} = 0.1387\pi$$

$$\Omega_c = \frac{\Omega_r}{\sqrt[2N]{(10^{A_r/10}-1)}} = \frac{0.3\pi}{\sqrt[6]{(10^{20/10}-1)}} = 0.1392\pi$$

选择 N=3，$\Omega_c = 0.1387\pi = 0.4357$

极点为 $p_1 = \Omega_c e^{j\frac{2}{3}\pi}$， $p_2 = \Omega_c e^{j\pi}$， $p_3 = \Omega_c e^{j\frac{4}{3}\pi}$

$H_a(s)$为

$$H_a(s) = \frac{(-1)^3 \Omega_c e^{j\frac{2}{3}\pi} \Omega_c e^{j\pi} \Omega_c e^{j\frac{4}{3}\pi}}{\left(s-\Omega_c e^{j\frac{2}{3}\pi}\right)(s-\Omega_c e^{j\pi})\left(s-\Omega_c e^{j\frac{4}{3}\pi}\right)} = \frac{\Omega_c^3}{s^3+2\Omega_c s^2+2\Omega_c^2 s+\Omega_c^3}$$

$$= \frac{0.0827}{s^3+0.8713s^2+0.3796s+0.0827}$$

在 MATLAB 中用 afd_butt(Omegap,Omegar,Ap,Ar)函数来设计巴特沃斯模拟低通滤波器，Ω 用 Omega 代替。

```
function[b,a]=afd_butt(Omegap,Omegar,Ap,Ar);
%巴特沃斯模拟低通滤波器的设计
%[b,a]=afd_butt(Omegap, Omegar,Ap,Ar);
%b=Ha(s)分子多项式的系数
%a=Ha(s)分母多项式的系数
%Omegap=通带截止频率,单位为弧度/秒; Omegap>0
%Omegar=阻带起始频率,单位为弧度/秒;Omegar>Omegap>0
%Ap=通带中幅度最大衰减的 dB 数
%Ar=阻带中幅度最小衰减的 dB 数
if Omegap<=0
   error('通带边缘必须大于 0')
end
if Omegar<=Omegap
   error('阻带边缘必须大于通带边缘')
end
N=ceil((log10((10^(0.1*Ar)-1)/(10^(0.1*Ap)-1)))/(2*log10(Omegar/Omegap)));
fprintf('\n***巴特沃斯模拟低通滤波器阶次=%2.0f\n',N);
OmegaC=Omegap/((10^(Ap/10)-1)^(1/(2*N)));
[b,a]=u_buttap(N,OmegaC);
```

MATLAB 提供了[z,p,k]=buttap(N)的函数，用来实现 N 阶归一化(Ω_c=1)巴特沃斯模拟低通滤波器原型，它返回数组零点 z 和极点 p 及增益 k。若设计非归一化($\Omega_c \neq 1$)巴特沃斯模拟低通滤波器原型，可用 u_buttap(N,OmegaC)函数进行变换。

```
function[b,a]=u_buttap(N,OmegaC);
%未归一化的巴特沃斯模拟低通滤波器原型
%[b,a]=u_buttap(N,OmegaC);
%b-Ha(s)分子多项式的系数
%a=Ha(s)分母多项式的系数
%N=滤波器的阶数
%OmegaC=截止频率，单位为 rad/s
[z,p,k]=buttap(N);
p=p*OmegaC;
k=k*OmegaC^N;
B=real(poly(z));
b0=k;
b=k*B;
a=real(poly(p));
```

为了显示模拟滤波器的频域图，可用 freqs_m(b,a,Omega_max)函数计算 $H_a(j\Omega)$的幅度响应和相位响应，其响应计算至最大频率Ω_{max}。

```
function[db,mag,pha,Omega]=freqs_m(b,a,Omega_max);
%s 域频率响应计算
Omega=[0:1:500]*Omega_max/500;
H=freqs(b,a,Omega);
mag=abs(H);
db=20*log10((mag+eps)/max(mag));
pha=angle(H);
```

用 sdir2cas 函数，可把直接形式转换成级联形式。

```
function[C,B,A]=sdir2cas(b,a)
%s 平面中直接型到级联型的转换
Na=length(a)-1;Nb=length(b)-1;
b0=b(1);b=b/b0;
a0=a(1);a=a/a0;
C=b0/a0;
p=cplxpair(roots(a));K=floor(Na/2);
if K*2==Na
   A=zeros(K,3);
   for n=1:2:Na
      Arow=p(n:1:n+1,:);Arow=poly(Arow);
      A(fix((n+1)/2),:)=real(Arow);
   end
 elseif Na==1
   A=[0 real(poly(p))];
 else
   A=zeros(K+1,3);
```

```
  for n=1:2:2*K
     Arow=p(n:1:n+1,:);Arow=poly(Arow);
     A(fix((n+1)/2),:)=real(Arow);
     end
  A(K+1,:)=[0 real(poly(p(Na)))];
end
z=cplxpair(roots(b));K=floor(Nb/2);
if Nb==0
  B=[0 0 poly(z)];
     elseif K*2==Nb
  B=zeros(K,3);
  for n=1:2:Nb
     Brow=z(n:1:n+1,:);Brow=poly(Brow);
     B(fix((n+1)/2),:)=real(Brow);
  end
 elseif Nb==1
  B=[0 real(poly(z))];
 else
   B=zeros(K+1,3);
  for n=1:2:2*K
     Brow=z(n:1:n+1,:);Brow=poly(Brow);
     B(fix((n+1)/2),:)=real(Brow);
     end
  B(K+1,:)=[0 real(poly(z(Nb)))];
 end
```

有了上面的函数，可实现例1的设计如下。

```
%s180.m
Omegap=0.2*pi;Omegar=0.3*pi;Ap=10;Ar=20;%技术指标
[b,a]=afd_butt(Omegap,Omegar,Ap,Ar) %巴特沃斯低通滤波器
[C,B,A]=sdir2cas(b,a) %计算二阶节系数,级联型实现
[db,mag,pha,Omega]=freqs_m(b,a,0.5*pi);%计算幅频响应
[ha,x,t]=impulse(b,a);%计算模拟滤波器的单位脉冲响应
subplot(221);plot(Omega/pi,mag);title('幅度响应|Ha(j\Omega)|');
subplot(222);plot(Omega/pi,db);title('幅度响应(dB)');
subplot(223);plot(Omega/pi,pha/pi);title('相位响应');
subplot(224);plot(t,ha);title('单位脉冲响应ha(t)');
axis([0,max(t),min(ha),max(ha)]);
```

运行后得到结果如图2-40所示。

```
b = 0.0827
a =1.0000    0.8713    0.3796    0.0827
C =0.0827
B =0     0     1
A =1.0000    0.4357    0.1898
      0    1.0000    0.4357
```

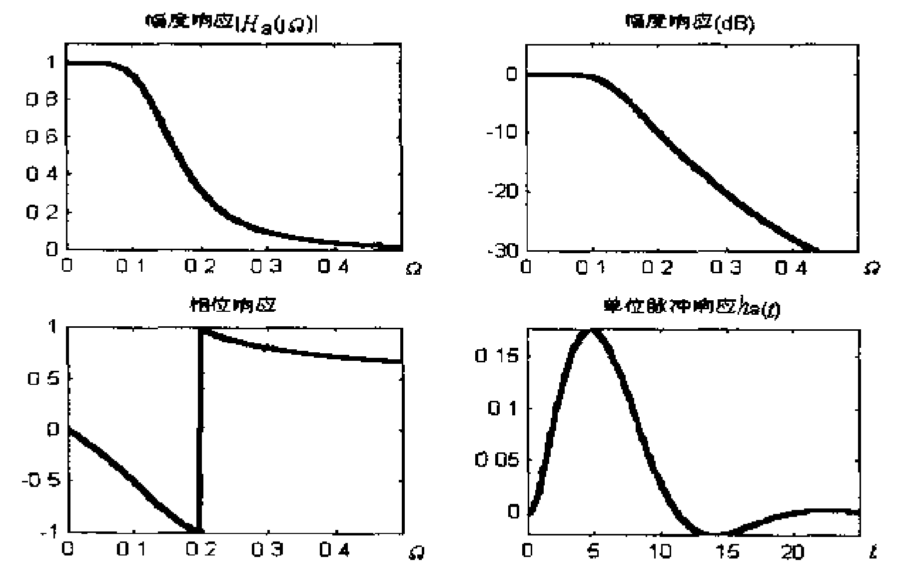


图 2-40　巴特沃斯模拟低通滤波器（例 1）

所以直接型
$$H_a(s)=\frac{0.0827}{s^3+0.8713s^2+0.3796s+0.0827}$$

级联型
$$H_a(s)=\frac{0.0827}{(s^2+0.4357s+0.1898)(s+0.4357)}$$

从图 2-40 可见这个滤波器通带截止频率处的衰减比较大（10dB），故所设计出的滤波器阶次较低（N=3）。

［例 2］　设计一个低通巴特沃斯模拟滤波器，要求

通带截止频率 $\Omega_p=0.2\pi$，通带最大衰减 $A_p=3\text{dB}$

阻带起始频率 $\Omega_r=0.3\pi$，阻带最小衰减 $A_r=20\text{dB}$

解：将例 1 的通带衰减变为 3dB，所设计的模拟低通滤波器系数为

b = 0.0617

a =1.0000　　2.4286　　2.9490　　2.2703　　1.1652　　0.3791　　0.0617

C =0.0617

B = 0　　0　　1

A = 1.0000　　1.2143　　0.3951

　　1.0000　　0.8889　　0.3951

　　1.0000　　0.3254　　0.3951

结果见图 2-41。

直接型

$$H_a(s)=\frac{0.0617}{s^6+2.4286s^5+2.9490s^4+2.2703s^3+1.1652s^2+0.3791s+0.0617}$$

级联型

$$H_a(s)=\frac{0.0617}{(s^2+1.2143s+0.3951)(s^2+0.8889s+0.3951)(s^2+0.3254s+0.3951)}$$

从图 2-40 可见这个滤波器通带截止频率处的衰减比较小（3dB），故所设计出的滤波器阶次较高（N=6），更接近理想低通滤波器。

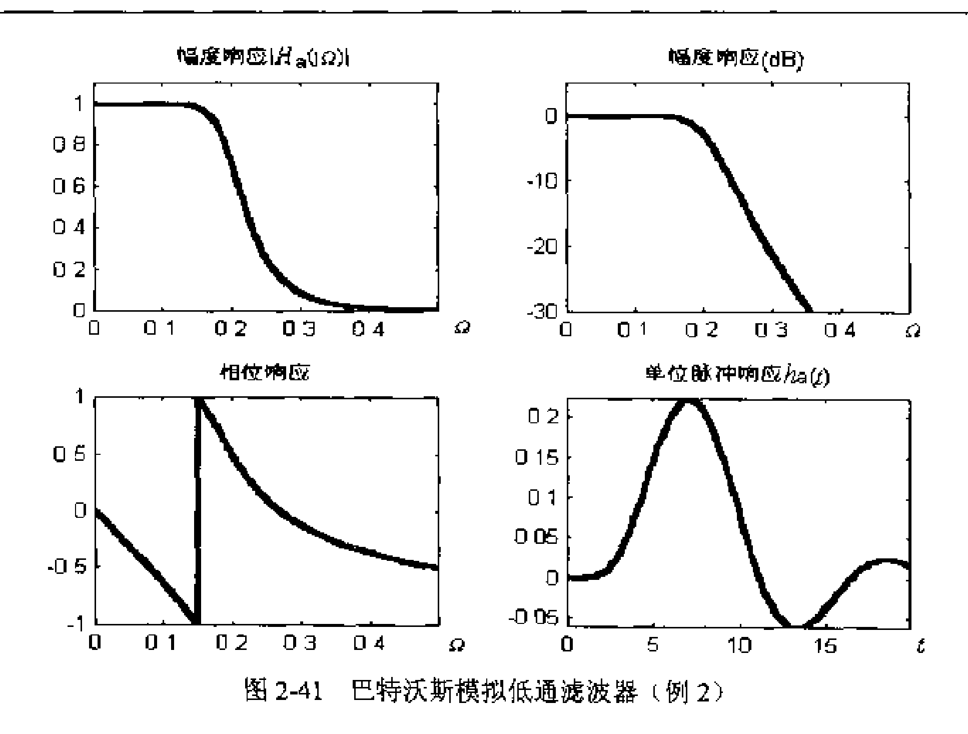


图 2-41 巴特沃斯模拟低通滤波器（例 2）

2.3.1.3 切比雪夫模拟滤波器

切比雪夫模拟滤波器的特性是：幅度具有等波纹波动的特性。在通带内是等波纹波动，在阻带内是单调的，称切比雪夫 1 型；在阻带内是等波纹波动，在通带内是单调的，称切比雪夫 2 型。

（1）切比雪夫 1 型模拟滤波器

切比雪夫 1 型模拟滤波器的特性和系统函数

切比雪夫 1 型模拟低通滤波器的幅度平方响应定义为

$$|H_a(j\Omega)|^2=\frac{1}{1+\varepsilon^2 C_N^2\left(\dfrac{\Omega}{\Omega_c}\right)}$$

式中 N ——滤波器的阶次；

ε ——通带波动系数，与通带波纹有关的一个参数；

Ω_c——截止频率，是滤波器的某一衰减分贝的通带宽度（切比雪夫滤波器中Ω_c不一定是 3dB 的带宽）。

$C_N(x)$ ——N 阶切比雪夫多项式。

表 2-1 切比雪夫多项式

N	$C_N(x)$
0	1
1	x
2	$2x^2-1$
3	$4x^3-3x$
4	$8x^4-8x^2+1$

切比雪夫多项式定义为：

$$C_N(x)=\begin{cases}\cos(N\cos^{-1}(x)), & 0\leqslant x\leqslant 1\\ \cosh(\cosh^{-1}(x)), & 1<x<\infty\end{cases}$$

其中 $x=\dfrac{\Omega}{\Omega_c}$。

可以展开成多项式，见表 2-1。

由此可归纳出高阶切比雪夫多项式的递推公式为

$$C_{N+1}(x)=2xC_N(x)-C_{N-1}(x)$$

切比雪夫模拟滤波器的等波动响应是由多项式 $C_N(x)$造成的。它的主要性质是

① $0<x\leqslant 1$ 时，$C_N(x)$在-1 和 1 之间出现振荡；

② $1<x<\infty$时，$C_N(x)$单调增至∞。

所以在 $0\leqslant\Omega\leqslant\Omega_c$时，$|H_a(j\Omega)|^2$的最大值为 1，最小值为$\dfrac{1}{1+\varepsilon^2}$；在$\Omega>\Omega_c$时，$|H_a(j\Omega)|^2$单调地趋近于零。

通带纹波表示为

$$\delta=20\lg\frac{|H_a(j\Omega)|_{max}}{|H_a(j\Omega)|_{min}}\quad(\text{dB})$$

这里$|H_a(j\Omega)|_{max}=1$，表示通带幅度响应的最大值；$|H_a(j\Omega)|_{min}=\dfrac{1}{\sqrt{1+\varepsilon^2}}$，表示通带幅度响应的最小值。故$\delta=10\lg(1+\varepsilon^2)$，因而

$$\varepsilon=\sqrt{10^{0.1\delta}-1}$$

用切比雪夫模拟滤波器的幅度平方响应左半平面的极点组成 $H_a(s)$

$$H_a(s)=\frac{K}{\prod_{i}^{N}(s-p_i)}$$

极点为

$$p_i=\sigma_i+j\Omega_i$$

其中

$$\left.\begin{aligned}\sigma_i&=-\Omega_c a\sin\left[\frac{\pi}{2N}(2i-1)\right],\\ \Omega_i&=\Omega_c b\cos\left[\frac{\pi}{2N}(2i-1)\right],\end{aligned}\right.\quad i=1,2,\cdots,N$$

$$a=\frac{1}{2}\left(\alpha^{\frac{1}{N}}-\alpha^{-\frac{1}{N}}\right)$$

$$b=\frac{1}{2}\left(\alpha^{\frac{1}{N}}+\alpha^{-\frac{1}{N}}\right)$$

$$\alpha=\frac{1}{\varepsilon}+\sqrt{\frac{1}{\varepsilon^2}+1}$$

K 为归一化因子，它使得

$$H_a(j0)=\begin{cases}1, & N\text{为奇数}\\ \dfrac{1}{\sqrt{1+\varepsilon^2}}, & N\text{为偶数}\end{cases}$$

切比雪夫 1 型模拟滤波器的设计方法

设计的基本思想是给定Ω_p、δ，并在Ω_r处满足指标A_r

$$\delta=10\lg[1+\varepsilon^2]$$

$$A_r=10\lg\left[1+\varepsilon^2 C_N^2\left(\frac{\Omega_r}{\Omega_c}\right)\right]$$

由上述两方程得到参数ε，阶次 N 和截止频率Ω_c

$$\varepsilon=\sqrt{10^{0.1\delta-1}}$$

而

$$\Omega_c=\Omega_p$$

令

$$A=10^{A_r/20}$$

$$\Omega_R=\frac{\Omega_r}{\Omega_p}$$

$$g=\sqrt{(A^2-1)/\varepsilon^2}$$

阶数 N 为

$$N\geqslant\frac{\lg[g+\sqrt{g^2-1}]}{\lg[\Omega_R+\sqrt{\Omega_R^2-1}]}$$

［例3］ 设计一个低通切比雪夫1型滤波器，满足

通带截止频率 $\Omega_p=0.2\pi$ rad/s，通带波动 $\delta=1$dB

阻带截止频率 $\Omega_r=0.3\pi$ rad/s，阻带衰减 $A_r=16$dB

解：首先计算需要的参数：

$$\varepsilon=\sqrt{10^{0.1\times1-1}}=0.5088$$

$$\Omega_c=\Omega_p=0.2\pi$$

$$A=10^{A_r/20}=10^{16/20}=6.3096$$

$$\Omega_R=\frac{\Omega_r}{\Omega_p}=\frac{0.3\pi}{0.2\pi}=1.5$$

$$g=\sqrt{(A^2-1)/\varepsilon^2}=\sqrt{(6.3096^2-1)/0.5088^2}=12.2429$$

滤波器的阶次

$$N\geqslant\frac{\lg[12.2429+\sqrt{12.2429^2-1}]}{\lg[1.5+\sqrt{1.5^2-1}]}=3.32$$

取 $N=4$

则

$$\alpha=\frac{1}{\varepsilon}+\sqrt{\frac{1}{\varepsilon^2}+1}=4.1702$$

$$a=\frac{1}{2}\left(\alpha^{\frac{1}{N}}-\alpha^{-\frac{1}{N}}\right)=0.3646$$

$$b=\frac{1}{2}\left(\alpha^{\frac{1}{N}}+\alpha^{-\frac{1}{N}}\right)=1.0644$$

$H_a(s)$在左半平面有4个极点

$$p_1=-a\Omega_c\sin\frac{\pi}{8}+\mathrm{j}b\Omega_c\cos\frac{\pi}{8}=-0.0877+\mathrm{j}0.6179$$

$$p_2=-a\Omega_c\sin\frac{3\pi}{8}+\mathrm{j}b\Omega_c\cos\frac{3\pi}{8}=-0.2117+\mathrm{j}0.2559$$

$$p_3=-a\Omega_c\sin\frac{5\pi}{8}+\mathrm{j}b\Omega_c\cos\frac{5\pi}{8}=-0.2117-\mathrm{j}0.2559$$

$$p_4 = -a\Omega_c \sin\frac{7\pi}{8} + jb\Omega_c \cos\frac{7\pi}{8} = -0.0877 - j0.6179$$

因此

$$H_a(s) = \frac{K}{\prod_{i}^{N}(s-p_i)} = \frac{0.03829}{(s^2+0.1754s+0.3895)(s^2+0.4234s+0.1103)}$$

MATLAB 提供了[z,p,k]=cheb1ap(N,Dt)的函数，用来实现 N 阶、通带波动为δ的归一化切比雪夫 1 型模拟滤波器，它返回数组零点 z 和极点 p 及增益 k。若设计非归一化切比雪夫 1 型模拟滤波器，可用 chb1ap(N,Dt,OmegaC)函数进行变换。

```
function[b,a]=u_chb1ap(N,Dt,OmegaC);
%未归一化的切比雪夫 1 型模拟低通滤波器原型
%b=Ha(s)分子多项式的系数
%a=Ha(s)分母多项式的系数
%OmegaC=截止频率, rad/s
%Dt=通带波动的+dB 数
%N=滤波器阶次
[z,p,k]=cheb1ap(N,Dt);
a=real(poly(p));
aNn=a(N+1);
p=p*OmegaC;
a=real(poly(p));
aNu=a(N+1);
k=k*aNu/aNn;
b0=k;
B=real(poly(z));
b=k*B;
```

以函数 u_chb1ap(N,Dt,OmegaC)为基础（根据N, δ, Ω_c确定切比雪夫 1 型滤波器），用上述方法编写函数 afd_chb1，根据给定指标设计切比雪夫 1 型模拟滤波器。

```
function[b,a]=afd_chb1(Omegap,Omegar,Dt,Ar);
%切比雪夫 1 型模拟低通滤波器设计
%b=Ha(s)分子的系数
%a=Ha(s)分母的系数
%Omegap=通带截止频率, rad/s
%Omegar=阻带起始频率, rad/s
%Dt=通带波动的+dB 数
%Ar=阻带最小衰减的+dB 数
if Omegap<=0
   error('通带边缘必须大于 0')
end
if Omegar<=Omegap
   error('阻带边缘必须大于通带边缘')
end
if(Dt<=0)|(Ar<0)
   error('通带波动或阻带衰减必须大于 0')
```

```
end
ep=sqrt(10^(Dt/10)-1);
A=10^(Ar/20);
OmegaC=Omegap;
OmegaR=Omegar/Omegap;
g=sqrt(A*A-1)/ep;
N=ceil(log10(g+sqrt(g*g-1))/log10(OmegaR+sqrt(OmegaR*OmegaR-1)));
fprintf('\n***切比雪夫 1 型模拟低通滤波器阶次=%2.0f\n',N);
[b,a]=u_chblap(N,Dt,OmegaC);
```

用 MATLAB 实现例 3 中的滤波器如下。

```
%s181.m
Omegap=0.2*pi;Omegar=0.3*pi;Dt=1;Ar=16;%技术指标
[b,a]=afd_chb1(Omegap,Omegar,Dt,Ar);%切比雪夫 1 型模拟低通滤波器
[C,B,A]=sdir2cas(b,a) %级联形式
[db,mag,pha,w]=freqs_m(b,a,pi);%计算幅频响应
[ha,x,t]=impulse(b,a);%计算模拟滤波器的单位脉冲响应
subplot(221);plot(w/pi,mag);title('幅度响应|Ha(j\Omega)|');
subplot(222);plot(w/pi,db);title('幅度响应(dB)');
subplot(223);plot(w/pi,pha/pi);title('相位响应');axis([0,1,-1,1]);
subplot(224);plot(t,ha);title('单位脉冲响应 ha(t)');
axis([0,max(t),min(ha),max(ha)]);
```

结果见图 2-42。

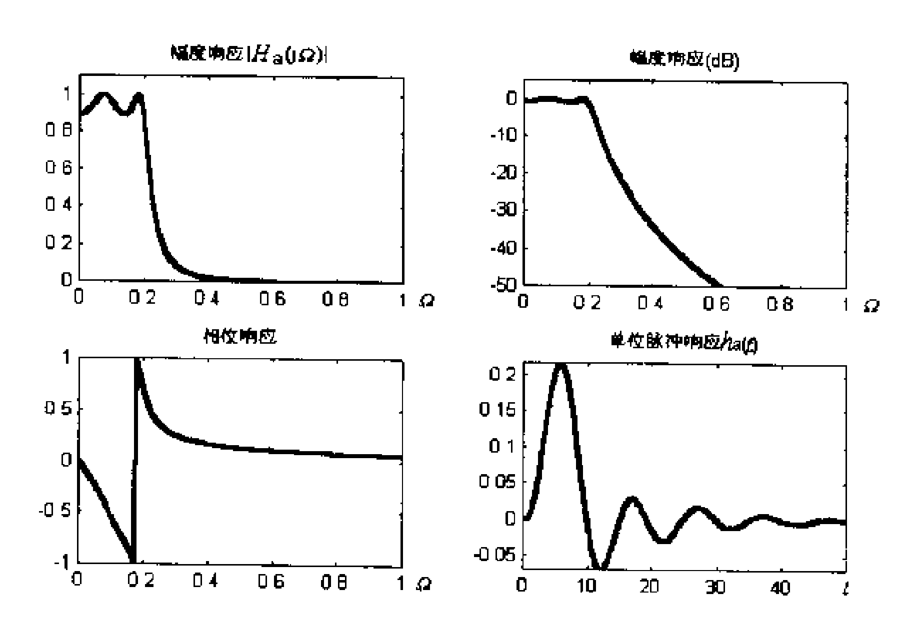


图 2-42 切比雪夫 1 型模拟低通滤波器

运行后得到 C =0.0383

B =0 0 1

A =1.0000 0.4233 0.1103

1.0000 0.1753 0.3895

4 阶切比雪夫 1 型模拟低通滤波器的系统函数为

$$H_a(s)=\frac{0.0383}{(s^2+0.4233s+0.1103)(s^2+0.1753s+0.3895)}$$

它所对应的频率响应曲线见图 2-42。

（2）切比雪夫 2 型模拟滤波器

切比雪夫 2 型模拟滤波器的特性和系统函数

切比雪夫 2 型模拟低通滤波器的幅度平方响应定义为

$$|H_a(j\Omega)|^2=\frac{1}{1+\left[\varepsilon^2 C_N^2\left(\frac{\Omega_c}{\Omega}\right)\right]^{-1}}$$

式中 N ——滤波器的阶次；

ε ——通带波动系数，与通带波纹有关的一个参数；

Ω_c——截止频率，是阻带衰减达到规定数值的最低频率（切比雪夫滤波器中Ω_c不一定是 3dB 的带宽）。

MATLAB 提供了[z,p,k]=cheb2ap(N,Ar)的函数，用来实现 N 阶、阻带波动为 Ar 的归一化切比雪夫 2 型模拟滤波器，它返回数组零点 z 和极点 p 及增益 k。若设计非归一化切比雪夫 2 型模拟滤波器，可用 chb2ap(N,Ar,OmegaC)函数进行变换。

```
function[b,a]=u_chb2ap(N,Ar,OmegaC);
%未归一化的切比雪夫 2 型模拟低通滤波器原型
%b=Ha(s)分子多项式的系数
%a=Ha(s)分母多项式的系数
%OmegaC=截止频率，rad/s
%Ar=阻带衰减的+dB 数
%N=滤波器阶次
[z,p,k]=cheb2ap(N,Ar);
      a=real(poly(p));
      aNn=a(N+1);
      p=p*OmegaC;
      a=real(poly(p));
      aNu=a(N+1);
      k=k*aNu/aNn;
      b0=k;
      B=real(poly(z));
      b=k*B;
```

以函数 u_chh2ap(N,Ar,OmegaC)为基础（根据 N，Ar，Ω_c确定切比雪夫 2 型滤波器），用函数 afd_chb2，根据给定指标设计切比雪夫 2 型模拟滤波器。

```
function[b,a]=afd_chb2(Omegap,Omegar,Dt,Ar);
%切比雪夫 2 型模拟低通滤波器设计
%b=Ha(s)分子的系数
%a=Ha(s)分母的系数
%Omegap=通带截止频率，rad/s
%Omegar=阻带起始频率，rad/s
%Dt=通带波动的+dB 数
```

```
%Ar=阻带衰减的+dB数
if Omegap<=0
   error('通带边缘必须大于 0')
end
if Omegar<=Omegap
   error('阻带边缘必须大于通带边缘')
end
if(Dt<=0)|(Ar<0)
   error('通带波动或阻带衰减必须大于 0')
end
ep=sqrt(10^(Dt/10)-1);
A=10^(Ar/20);
OmegaC=Omegar;
OmegaR=Omegar/Omegap;
g=sqrt(A*A-1)/ep;
N=ceil(log10(g+sqrt(g*g-1))/log10(OmegaR+sqrt(OmegaR*OmegaR-1)));
fprintf('\n***切比雪夫2型模拟低通滤波器阶次=%2.0f\n',N);
[b,a]=u_chb2ap(N,Ar,OmegaC);
```

[例 4] 设计一个切比雪夫 2 型低通滤波器，满足

通带截止频率 $\Omega_p = 0.2\pi$ rad/s，通带波动δ=1dB

阻带截止频率 $\Omega_r = 0.3\pi$ rad/s，阻带衰减 A_r=16dB

```
%s182.m
Omegap=0.2*pi;Omegar=0.3*pi;Dt=1;Ar=16;
[b,a]=afd_chb2(Omegap,Omegar,Dt,Ar);
[C,B,A]=sdir2cas(b,a)
[db,mag,pha,w]=freqs_m(b,a,pi);
[ha,x,t]=impulse(b,a);
subplot(221);plot(w/pi,mag);title('幅度响应|Ha(j\Omega)|');
subplot(222);plot(w/pi,db);title('幅度响应(dB)');
subplot(223);plot(w/pi,pha/pi);title('相位响应');
subplot(224);plot(t,ha);title('单位脉冲响应ha(t)');
axis([0,max(t),min(ha),max(ha)]);
```

运行后得到 C = 0.1585

```
        B = 1.0000         0        6.0654
            1.0000         0        1.0407
        A = 1.0000    1.9521        1.4747
            1.0000    0.3719        0.6784
```

4 阶切比雪夫 2 型模拟低通滤波器的系统函数为：

$$H_a(s) = \frac{0.1585(s^2 + 6.0654)(s^2 + 1.0407)}{(s^2 + 1.9521s + 1.4747)(s^2 + 0.3719s + 0.6784)}$$

它所对应的频率响应曲线如图 2-43 所示。

从图 2-43 可见，在通带中 2 型比 1 型切比雪夫模拟低通滤波器的相位响应更线性。

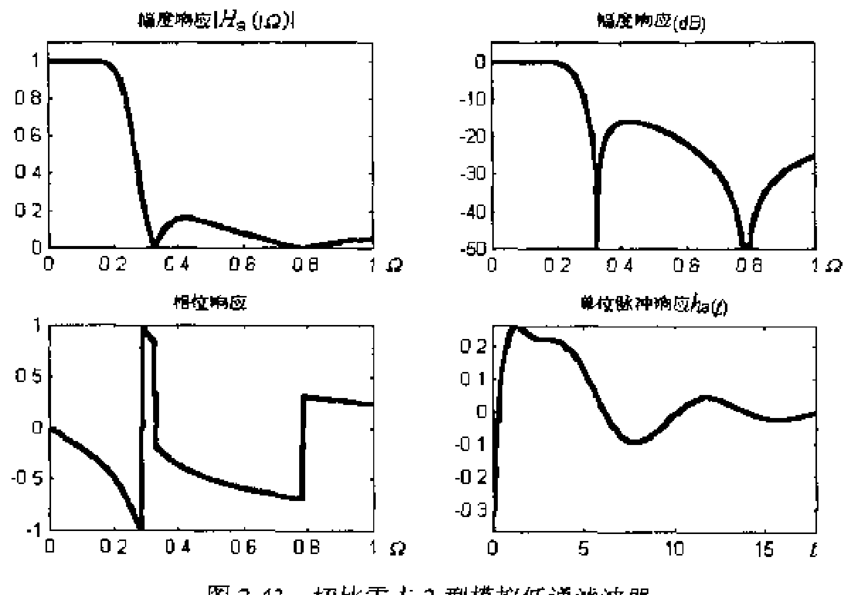


图 2-43 切比雪夫 2 型模拟低通滤波器

2.3.2 模拟到数字滤波器的设计

设计数字滤波器可以按技术要求设计一个模拟滤波器 $H_a(s)$，再按一定的转换关系，将 $H_a(s)$转换成数字滤波器的 $H(z)$。这样问题就变成如何将 s 平面上的 $H_a(s)$转换成 z 平面上的数字滤波器 $H(z)$。为了保证转换后的 $H(z)$仍满足技术要求，必须对转换关系提出如下要求：

① 为了保证因果稳定关系，转换关系应使 s 平面的左半平面映射 z 平面的单位圆内部；

② 数字滤波器的频响应模仿 $H_a(s)$的频响，转换关系应使 s 平面的虚轴映射到 z 平面的单位圆上。

将 $H_a(s)$从 s 平面上转换到 z 平面的 $H(z)$有多种方法，但在工程上常采用脉冲响应不变法和双线性变换法。

2.3.2.1 脉冲响应不变法

（1）基本原理

设模拟滤波器为 $H_a(s)$，相应的单位脉冲响应为 $h_a(t)$，希望数字滤波器的单位脉冲响应 $h(n)$ 与 $h_a(t)$相似。对 $h_a(t)$以采样间隔 T 对其进行采样，得到

$$h(n)=h_a(t)|_{t=nT}=h_a(nT)$$

参数 T 的选择应使样本能"捕获" $h_a(t)$的形状。由于这是一个采样操作，则模拟和数字频率的关系为

$$\omega=\Omega T$$

（2）混叠误差

由于在单位圆上 $z=\mathrm{e}^{\mathrm{j}\omega}$，在虚轴上 $s=\mathrm{j}\Omega$，因此从 s 平面到 z 平面的变换为

$$z=\mathrm{e}^{sT} \tag{2-4}$$

根据

$$H(\mathrm{e}^{\mathrm{j}\omega})=\frac{1}{T}\sum_{m=-\infty}^{\infty}H_a\left[\mathrm{j}\left(\frac{\omega}{T}-\frac{2\pi}{T}m\right)\right]$$

可得 $H(z)$和 $H_a(s)$的关系为

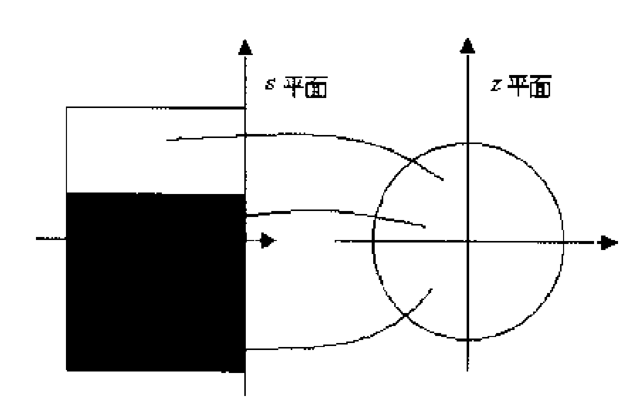


图 2-44 脉冲响应不变法的映射关系

$$H(z)=\frac{1}{T}\sum_{m=-\infty}^{\infty}H_a\left(s-j\frac{2\pi}{T}m\right)$$

经过式（2-4）的映射，复平面的变换如图 2-44 所示，从图 2-44 可见：

① 设$s=\sigma+j\Omega$，则

$\sigma<0$，映射为$|z|<1$（单位圆内）

$\sigma=0$，映射为$|z|=1$（单位圆上）

$\sigma>0$，映射为$|z|>1$（单位圆外）

② 所有宽度为$\frac{2\pi}{T}$的左半无限带的映射均为$|z|<1$，因此这个映射不是唯一的，而是多对一的映射。

③ 由于平面的左半平面映射为单位圆内，因此一个因果的稳定的模拟滤波器映射成因果的稳定的数字滤波器。

④ $|\Omega|\geqslant\frac{\pi}{T}$时，若$H_a(j\Omega)=H_a\left(j\frac{\omega}{T}\right)=0$，则

$$H(e^{j\omega})=\frac{1}{T}H_a\left(j\frac{\omega}{T}\right),\quad |\omega|\leqslant\pi$$

不存在混叠。但是，有限阶数的模拟滤波器的带宽不可能是真正有限的。因此，脉冲响应不变法的设计过程会出现混叠误差，而采样间隔 T 的影响是次要的。$H(e^{j\omega})$的幅度特性与采样间隔成反比，T较小时，$H(e^{j\omega})$的增益会很高。为避免这一现象，令$h(n)=Th_a(nT)$。

（3）设计过程

给定数字低通滤波器技术指标ω_p, ω_r, A_p, A_r，先设计一个等价的模拟低通滤波器，然后把它映射成希望的数字滤波器。

① 确定T并选择模拟频率

$$\Omega_p=\frac{\omega_p}{T}\quad 和\quad \Omega_r=\frac{\omega_r}{T}$$

② 根据指标ω_p, ω_r, A_p, A_r，设计模拟低通滤波器$H_a(s)$（巴特沃斯、切比雪夫等）。

③ 利用部分分式展开式，把$H_a(s)$展成

$$H_a(s)=\sum_{k=1}^{N}\frac{R_k}{s-p_k}$$

④ 把模拟极点p_k转换成数字极点e^{p_kT}，得到数字滤波器

$$H(z)=\sum_{k=1}^{N}\frac{TR_k}{1-e^{p_kT}z^{-1}}$$

已知模拟低通滤波器的$H_a(s)$为

$$H_a(s)=\frac{0.5012}{s^2+0.6449s+0.7079}$$

用脉冲响应不变法将$H_a(s)$转换成数字滤波器的$H(z)$。

解：把$H_a(s)$展成

$$H_a(s)=\frac{-j0.3224}{s+0.3224+j0.7772}+\frac{j0.3224}{s+0.3224-j0.7772}$$

$H_a(s)$的极点为

$$p_1 = -(0.3224 + \mathrm{j}0.7772)$$
$$p_2 = -(0.3224 - \mathrm{j}0.7772)$$

$H(z)$的极点为 $z_1 = \mathrm{e}^{p_1 T}, z_2 = \mathrm{e}^{p_2 T}$

则脉冲响应不变法所设计的数字滤波器为

$$H(z) = \frac{-2\mathrm{e}^{-0.3224T} \times 0.3224T\sin(0.7772T)z^{-1}}{1-2z^{-1}\mathrm{e}^{-0.3224T} \times \cos(0.7772T) + \mathrm{e}^{-0.6449T}z^{-2}}$$

当 T=1s 时， $H_1(z) = \dfrac{-0.3276z^{-1}}{1-1.0329z^{-1}+0.5247z^{-2}}$

T=0.1s 时， $H_2(z) = \dfrac{-0.0048z^{-1}}{1-1.9307z^{-1}+0.9375z^{-2}}$

用 MATLAB 函数 imp_invr 实现脉冲响应不变法。已知有理函数 $H_a(s)$，利用 residue 函数可以得到它的零极点，然后把每一个模拟极点映射成数字极点。最后函数 residuez 把 $H(z)$ 转换成有理形式。

```
function[b,a]=imp_invr(c,d,T)
%从模拟滤波器到数字滤波器的脉冲响应不变变换
%b=数字滤波器的自变量为 z^-1 的分子多项式
%a=数字滤波器的自变量为 z^-1 的分母多项式
%c=模拟滤波器的自变量为 s 的分子多项式
%d=模拟滤波器的自变量为 s 的分母多项式
%T=采样间隔
[R,p,k]=residue(c,d);%模拟极点
p=exp(p*T);%数字极点
[b,a]=residuez(R,p,k);%把 H(z)转换成有理形式
b=real(b).*T;a=real(a);
```

用 imp_invr 函数实现例 1 的数字滤波器，并画出幅度特性曲线。

```
%s183.m
c=[0.5012];d=[1,0.6449,0.7079];T=1;
[b,a]=imp_invr(c,d,T)
[db,mag,pha,Omega]=freqs_m(c,d,2*pi);
[db,mag,pha,w]=freqz_m(b,a);
subplot(111);plot(Omega/pi,db,w/pi,db);
```

运行后得到如下结果。

```
b =0          0.3276
a =1.0000   -1.0329     0.5247
```

T=1 时的数字滤波器为 $H_1(z) = \dfrac{-0.3276z^{-1}}{1-1.0329z^{-1}+0.5247z^{-2}}$

T=0.1 时的数字滤波器为 $H_2(z) = \dfrac{-0.0048z^{-1}}{1-1.9307z^{-1}+0.9375z^{-2}}$

将 $H_a(\mathrm{j}\Omega)$、$H_1(\mathrm{e}^{\mathrm{j}\omega})$、$H_2(\mathrm{e}^{\mathrm{j}\omega})$ 的幅度特性分别用它们的最大值归一化，用 dB 表示，如图 2-45 所示。由图可见，模拟滤波器和数字滤波器幅度差异主要在π附近，当 T=1s 时，差异较大，主要是由于频率混叠现象造成的，当 T=0.1s 时差异大大减小。当 $\omega > \pi$ 后，曲线上升，这是由于 $H(\mathrm{e}^{\mathrm{j}\omega})$ 的周期性形成的。

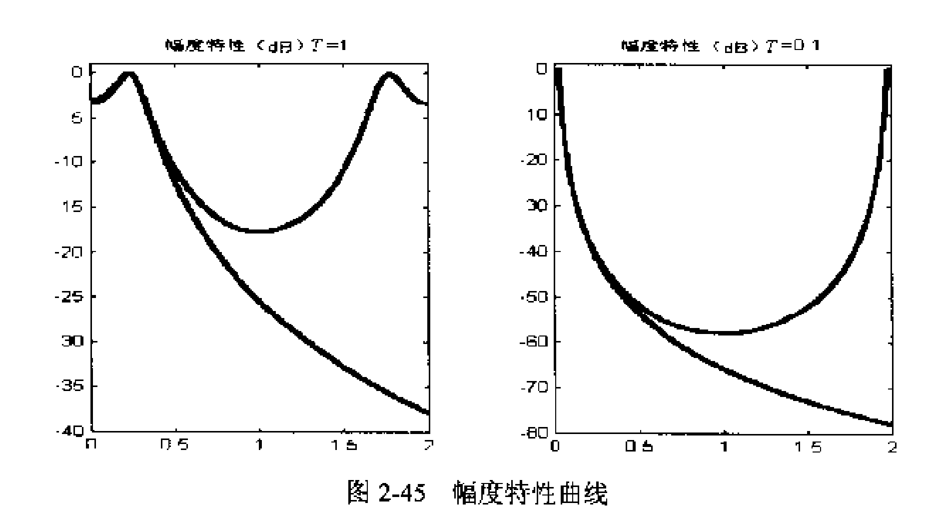


图 2-45　幅度特性曲线

［例 5］　设计低通数字滤波器，要求在通带内频率低于 0.2π rad 时，允许幅度误差在 1dB 以内，在频率 0.3π rad 到π rad 之间的阻带衰减大于 15dB。用脉冲响应不变法设计数字滤波器。T=1。①模拟滤波器采用巴特沃斯滤波器原型；②模拟滤波器采用切比雪夫滤波器原型。

解：① 模拟滤波器采用巴特沃斯滤波器原型。

$$\Omega_{p}=\frac{\omega_{p}}{T}=\frac{0.2\pi}{T}=0.2\pi$$

$$\Omega_{r}=\frac{\omega_{r}}{T}=\frac{0.3\pi}{T}=0.3\pi$$

$$N\geqslant\frac{\lg[(10^{A_p/10}-1)/(10^{A_r/10}-1)]}{2\lg(\Omega_{p}/\Omega_{r})}=5.884$$

取阶数 N=6

$$\Omega_{c}=\frac{\Omega_{p}}{\sqrt[2N]{(10^{A_p/10}-1)}}=0.7032\text{rad/s}$$

则

$$H_{a}(s)=\frac{0.0383}{s^{4}+0.5987s^{3}+0.5740s^{2}+0.1842s+0.0430}$$

将 $H_a(s)$展成部分分式，并作变换得到

$$H(z)=\frac{1.8557-0.6304z^{-1}}{1-0.9973z^{-1}+0.2570z^{-2}}+\frac{-2.1428+1.1454z^{-1}}{1-1.0691z^{-1}+0.3699z^{-2}}+\frac{0.2871-0.4466z^{-1}}{1-1.2972z^{-1}+0.6949z^{-2}}$$

```
%s184.m
wp=0.2*pi;wr=0.3*pi;Ap=1;Ar=15;
T=1;Omegap=wp/T;Omegar=wr/T;
[cs,ds]=afd_butt(Omegap,Omegar,Ap,Ar) %巴特沃斯原型
[C,B,A]=sdir2cas(cs,ds);%模拟滤波器级联型
[db,mag,pha,Omega]=freqs_m(cs,ds,pi);%模拟滤波器响应
subplot(224);plot(Omega/pi,mag);
title('模拟滤波器幅度响应|Ha(j\Omega)|');
[b,a]=imp_invr(cs,ds,T);%脉冲响应不变法设计
[C,B,A]=dir2par(b,a) %数字低通滤波器并联型
```

```
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(221);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)| ');
subplot(222);plot(w/pi,db);title('数字滤波器幅度响应 (dB)');
subplot(223);plot(w/pi,pha/pi);title('数字滤波器相位响应');
```

运行后得到

巴特沃斯模拟低通滤波器阶次=6

cs =0.1209

ds =1.0000　2.7170　3.6910　3.1788　1.8252　0.6644　0.1209

C =[]

B = 1.8557　-0.6304
　-2.1428　1.1454
　0.2871　-0.4466

A =1.0000　-0.9973　0.2570
　1.0000　-1.0691　0.3699
　1.0000　-1.2972　0.6949

巴特沃斯模拟滤波器为

$$H_a(s)=\frac{0.1209}{s^6+2.7170s^5+3.6910s^4+3.1788s^3+1.8252s^2+0.6644s+0.1209}$$

巴特沃斯数字滤波器为

$$H(z)=\frac{1.8557-0.6304z^{-1}}{1-0.9973z^{-1}+0.2570z^{-2}}+\frac{-2.1428+1.1454z^{-1}}{1-1.0691z^{-1}+0.3699z^{-2}}+\frac{0.2871-0.4466z^{-1}}{1-1.2972z^{-1}+0.6949z^{-2}}$$

脉冲响应不变法适合用并联型结构实现，从图 2-46 可见所设计的滤波器满足技术指标。

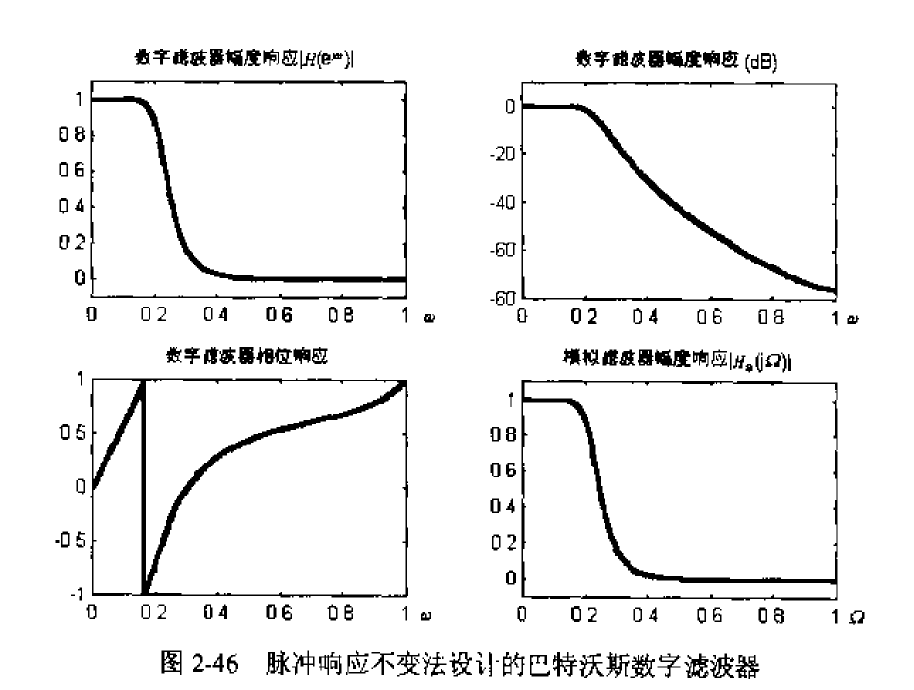


图 2-46　脉冲响应不变法设计的巴特沃斯数字滤波器

② 模拟滤波器采用切比雪夫滤波器原型

```
%s185.m
wp=0.2*pi;wr=0.3*pi;Ap=1;Ar=15;
```

```
T=1;Omegap=wp/T;Omegar=wr/T;
[cs,ds]=afd_chbl(Omegap,Omegar,Ap,Ar) %切比雪夫 1 原型
[C,B,A]=sdir2cas(cs,ds);%模拟滤波器级联型
[db,mag,pha,Omega]=freqs_m(cs,ds,pi);%模拟滤波器响应
subplot(224);plot(Omega/pi,mag);
title('模拟滤波器幅度响应|Ha(j\Omega)|');
[b,a]=imp_invr(cs,ds,T);%脉冲响应不变法设计
[C,B,A]=dir2par(b,a) %数字低通滤波器并联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(221);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(222);plot(w/pi,db);title('数字滤波器幅度响应 (dB)');
subplot(223);plot(w/pi,pha/pi);title('数字滤波器相位响应');
```

运行后得到

切比雪夫 1 型模拟低通滤波器阶次=4

cs =0.0383

ds =1.0000 0.5987 0.5740 0.1842 0.0430

C =[]

B =-0.0833 −0.0246

0.0833 0.0239

A =1.0000 −1.4934 0.8392

1.0000 −1.5658 0.6549

切比雪夫 1 型模拟滤波器为

$$H_a(s)=\frac{0.0383}{s^4+0.5987s^3+0.5740s^2+0.1842s+0.0430}$$

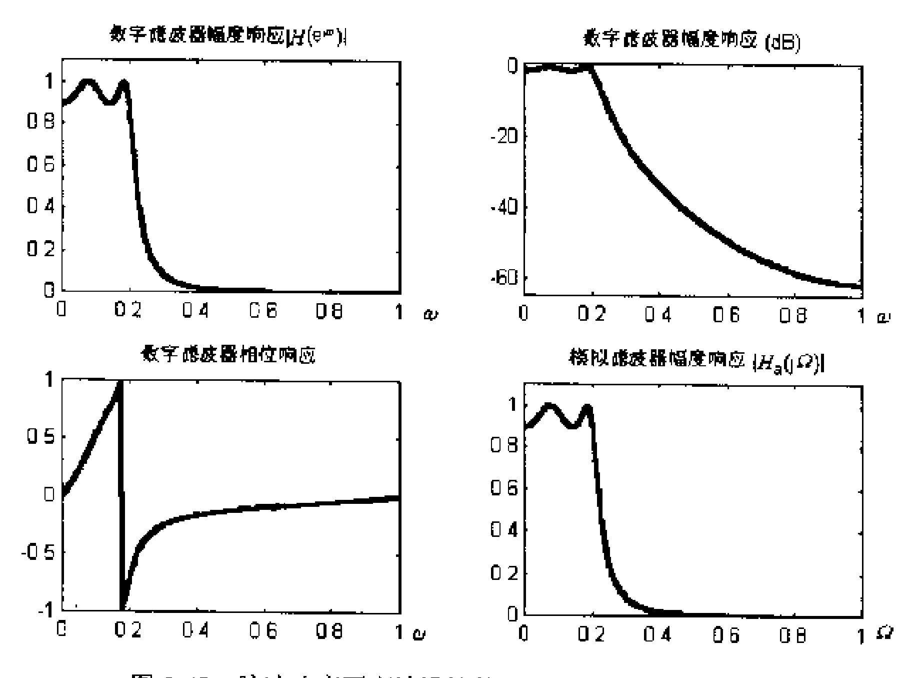


图 2-47 脉冲响应不变法设计的切比雪夫 1 型数字滤波器

切比雪夫 1 型数字滤波器为

$$H(z)=\frac{-0.0833-0.0246z^{-1}}{1-1.4934z^{-1}+0.8392z^{-2}}+\frac{0.0833+0.0239z^{-1}}{1-1.5658z^{-1}+0.6549z^{-2}}$$

从图 2-47 可见，所设计的滤波器满足技术指标。

当 T=0.5 时，切比雪夫 1 型模拟滤波器为

$$H_a(s)=\frac{0.6162}{s^4+1.1973s^3+2.2959s^2+1.4737s+0.6873}$$

当 T=2 时，切比雪夫 1 型模拟滤波器为

$$H_a(s)=\frac{0.0024}{s^4+0.2993s^3+0.1435s^2+0.0230s+0.0027}$$

不同 T 时，模仿的模拟滤波器不同，但得到的数字滤波器是相同的，结果见图 2-48。

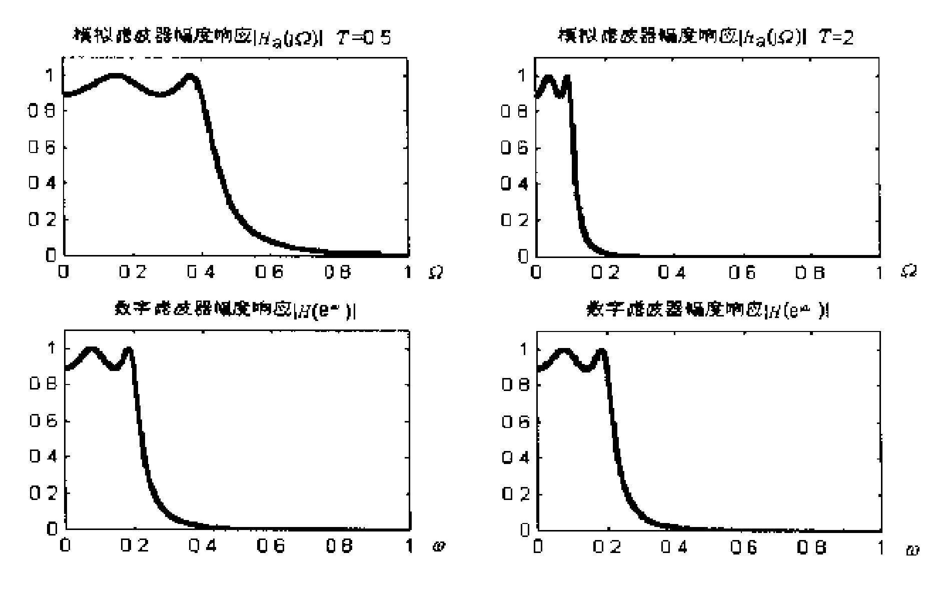


图 2-48　不同 T 时的滤波器

切比雪夫 2 型模拟低通滤波器阶次=4

cs = 0.1778　　0　　1.2637　　0　　1.1225

ds = 1.0000　　2.3696　　3.0322　　1.9925　　1.1225

C = 0.1778

B = –0.0313　　0.5760

　　–0.3901　　0.1552

A = 1.0000　　–0.5273　　0.1348

　　1.0000　　–1.1386　　0.6938

当模拟滤波器采用切比雪夫 2 型时，切比雪夫 2 型模拟滤波器为

$$H_a(s)=\frac{0.1778s^4+1.2637s^2+1.1225}{s^4+2.3696s^3+3.0322s^2+1.9925s+1.1225}$$

切比雪夫 2 型数字滤波器为

$$H(z)=\frac{-0.0313+0.5760z^{-1}}{1-0.5273z^{-1}+0.1348z^{-2}}+\frac{-0.3901+0.1552z^{-1}}{1-1.1386z^{-1}+0.6938z^{-2}}+0.1778$$

结果见图 2-49。

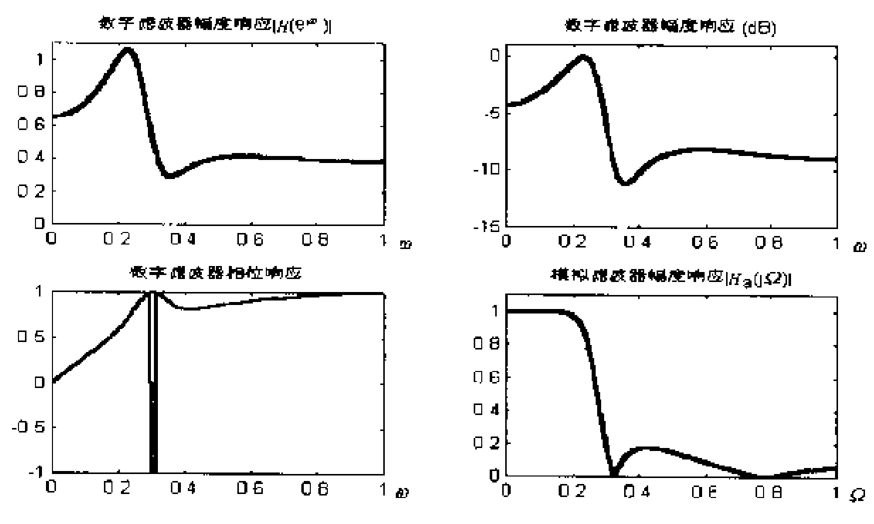


图 2-49 脉冲响应不变法设计的切比雪夫 2 型数字滤波器

从图 2-49 可见，所设计的滤波器不满足技术指标。这是因为切比雪夫 2 型滤波器在阻带是等波动的，它在阻带中高频段的响应不趋于零，所以经过脉冲响应不变法，其混叠效应很明显，降低通带和阻带的响应。

脉冲响应不变法的优点是频率Ω和ω是线性关系，缺点是频率响应有混叠，在某些情况下这种混叠是不可接受的。只有当模拟滤波器基本上是有限带宽的低通或带通滤波器，而且在阻带中不存在振荡时，这种方法才是有效的。

2.3.2.2 双线性变换法

（1）基本原理

为了克服多值映射的缺点，首先把整个 s 平面压缩变换到 s_1 平面的一条横带里，然后用标准变换关系 $z=e^{s_1T}$，将此横带变换到整个 z 平面，这样使得 s 平面与 z 平面是一一对应的关系，消除了多值变换性，达到消除混叠的目的，如图 2-50 所示。

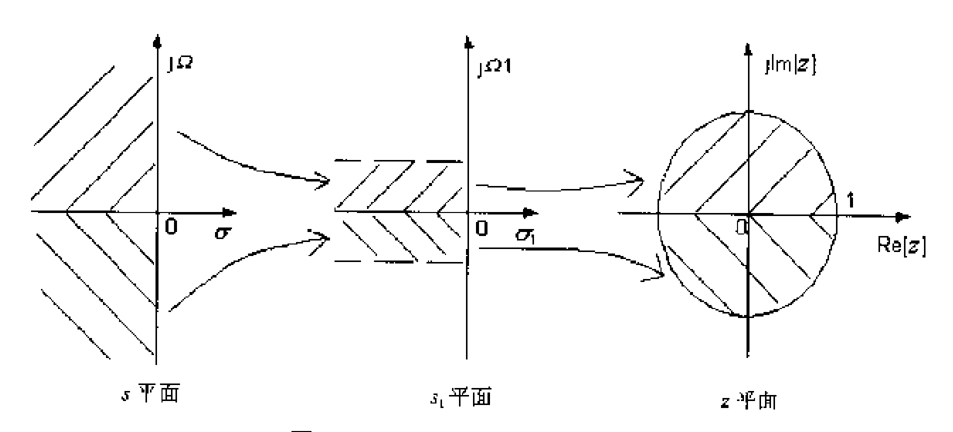


图 2-50 双线性变换的映射关系

这种映射用变换

$$s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \quad \Rightarrow z=\frac{1+sT/2}{1-sT/2}$$

其中 T 是变换参数，这种变换叫双线性变换。这是因为

$$\frac{T}{2}sz+\frac{T}{2}s-z+1=0$$

若固定其中一个变量，则另一个是线性的。或者说 s 和 z 是双线性的。

① 设 $s=\sigma+\mathrm{j}\Omega$，则

$$z=\frac{1+sT/2}{1-sT/2}=\frac{1+\frac{\sigma T}{2}+\mathrm{j}\frac{\Omega T}{2}}{1-\frac{\sigma T}{2}-\mathrm{j}\frac{\Omega T}{2}}$$

$\sigma<0$，映射为 $|z|<1$（单位圆内）

$\sigma=0$，映射为 $|z|=1$（单位圆上）

$\sigma>0$，映射为 $|z|>1$（单位圆外）

② 整个左半平面映射成单位圆内部，是一个稳定的变换。

③ 虚轴以一一对应的方式映射成单位圆，因此在频域不会出现混叠。

④ 频率变换关系 $\Omega=\frac{2}{T}\tan\frac{\omega}{2}$。

（2）设计过程

给定数字滤波器技术指标 ω_{p}, ω_{r}, A_{p}, A_{r}，先设计一个等价的模拟低通滤波器，然后把它变换成希望的数字滤波器。

① 确定 T，它是任意的，也可置为 1。

② 对截止频率 ω_{p}, ω_{r} 进行预畸

$$\Omega_{\mathrm{p}}=\frac{2}{T}\tan\frac{\omega_{\mathrm{p}}}{2} \quad 和 \quad \Omega_{\mathrm{r}}=\frac{2}{T}\tan\frac{\omega_{\mathrm{r}}}{2}$$

③ 设计模拟低通滤波器 $H_{\mathrm{a}}(s)$，使其满足指标 Ω_{p}, Ω_{r}, A_{p}, A_{r}（巴特沃斯、切比雪夫等）。

④ 将模拟滤波器变换成数字滤波器

$$H(z)=H_{\mathrm{a}}(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

MATLAB 提供 bilinear 函数实现双线性变换。

[例 6]　设计低通数字滤波器，要求在通带内频率低于 0.2π rad 时，允许幅度误差在 1dB 以内，在频率 0.3π rad 到π rad 之间的阻带衰减大于 15dB。用双线性变换法设计数字滤波器。（1）模拟滤波器采用巴特沃斯滤波器原型；（2）模拟滤波器采用切比雪夫滤波器原型。

解：（1）模拟滤波器采用巴特沃斯滤波器原型

选择 T=1

预畸

$$\Omega_{\mathrm{p}}=\frac{2}{T}\tan\frac{\omega_{\mathrm{p}}}{2}=2\tan\frac{0.2\pi}{2}=2\tan 0.1\pi$$

$$\Omega_{\mathrm{r}}=\frac{2}{T}\tan\frac{\omega_{\mathrm{r}}}{2}=2\tan\frac{0.3\pi}{2}=2\tan 0.15\pi$$

求模拟巴特沃斯滤波器

$$N \geqslant \frac{\lg[(10^{A_p/10}-1)/(10^{A_r/10}-1)]}{2\lg(\Omega_p/\Omega_r)} = 5.306$$

为满足指标，取阶数 N=6。

$$\Omega_c = \frac{\Omega_p}{\sqrt[2N]{(10^{A_p/10}-1)}} = 0.7273\text{rad/s}$$

则 $$H_a(s) = \frac{0.1480}{(s^2+1.4050s+0.5290)(s^2+1.0285s+0.5290)(s^2+0.3765s+0.5290)}$$

用双线性变换法求数字滤波器

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{0.00057969(1+z^{-1})^6}{(1-0.9459z^{-1}+0.2342z^{-2})(1-1.0541z^{-1}+0.3753z^{-2})(1-1.3143z^{-1}+0.7149z^{-2})}$$

```
%s186.m
wp=0.2*pi;wr=0.3*pi;Ap=1;Ar=15;
T=1;Omegap=(2/T)*tan(wp/2);Omegar=(2/T)*tan(wr/2);
[cs,ds]=afd_butt(Omegap,Omegar,Ap,Ar) %巴特沃斯原型
[C,B,A]=sdir2cas(cs,ds);%模拟滤波器级联型
[db,mag,pha,Omega]=freqs_m(cs,ds,pi);%模拟滤波器响应
subplot(224);plot(Omega/pi,mag);
title('模拟滤波器幅度响应|Ha(j\Omega)|');
[b,a]= bilinear(cs,ds,T);%双线性变换法设计
[C,B,A]=dir2cas(b,a) %数字低通滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(221);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)  ');
subplot(222);plot(w/pi,db);title('数字滤波器幅度响应 (dB)');
subplot(223);plot(w/pi,pha/pi);title('数字滤波器相位响应');
delta_w=2*pi/1000;
Ap=-(min(db(1:1:wp/delta_w+1)))
Ar=-round(max(db(wr/delta_w+1:1:501)))
```

运行后得到结果如图 2-51 所示。

巴特沃斯模拟低通滤波器阶次=6

C = 0.1480

B = 0 0 1

A =1.0000 1.4050 0.5290
1.0000 1.0285 0.5290
1.0000 0.3765 0.5290

cs =0.1480

巴特沃斯模拟低通滤波器为

$$H_a(s)=\frac{0.1480}{(s^2+1.4050s+0.5290)(s^2+1.0285s+0.5290)(s^2+0.3765s+0.5290)}$$

C = 5.7969e–004

B = 1.0000　2.0183　1.0187
　　1.0000　1.9814　0.9817
　　1.0000　2.0003　1.0000

A = 1.0000　–0.9459　0.2342
　　1.0000　–1.0541　0.3753
　　1.0000　–1.3143　0.7149

巴特沃斯数字低通滤波器为

$$H(z)=\frac{0.00057969(1+2.0182z^{-1}+1.0187z^{-2})}{(1-0.9459z^{-1}+0.2342z^{-2})}\times$$
$$\frac{(1+1.9814z^{-1}+0.9817z^{-2})(1+2.0003z^{-1}+1.0000z^{-2})}{(1-1.0541z^{-1}+0.3753z^{-2})(1-1.3143z^{-1}+0.7149z^{-2})}$$

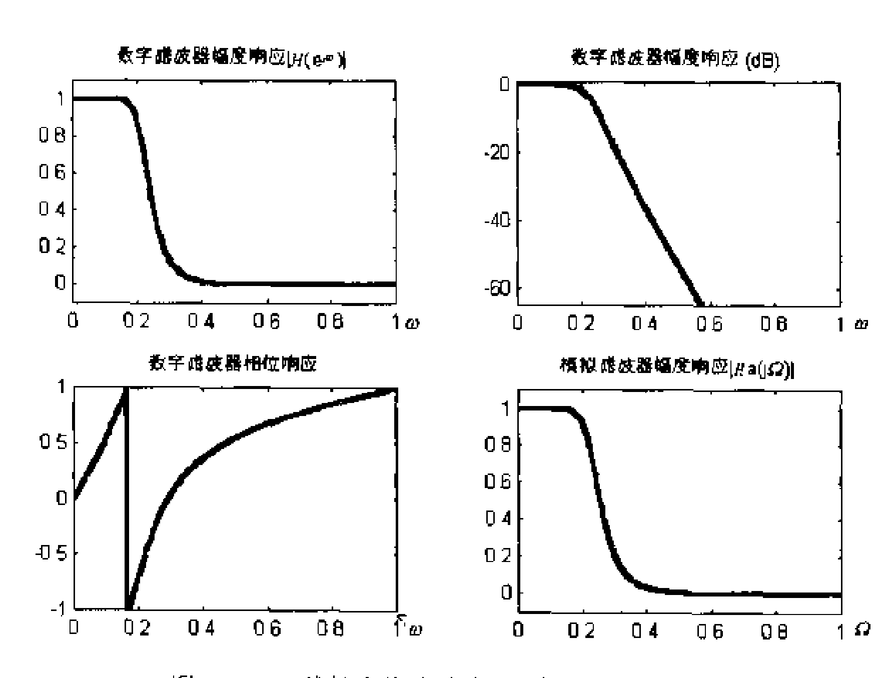


图 2-51　双线性变换法设计的巴特沃斯数字滤波器

（2）模拟滤波器采用切比雪夫滤波器原型

```
%s187.m
wp=0.2*pi;wr=0.3*pi;Ap=1;Ar=15;
T=1;Omegap=(2/T)*tan(wp/2);Omegar=(2/T)*tan(wr/2);
[cs,ds]=afd_chb1(Omegap,Omegar,Ap,Ar)  %切比雪夫原型
[C,B,A]=sdir2cas(cs,ds);%模拟滤波器级联型
[db,mag,pha,Omega]=freqs_m(cs,ds,pi);%模拟滤波器响应
subplot(224);plot(Omega/pi,mag);
title('模拟滤波器幅度响应|Ha(j\Omega)|');
[b,a]= bilinear(cs,ds,T);%双线性变换法设计
[C,B,A]=dir2cas(b,a)  %数字低通滤波器级联型
```

```
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(221);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(222);plot(w/pi,db);title('数字滤波器幅度响应 (dB)');
subplot(223);plot(w/pi,pha/pi);title('数字滤波器相位响应');
delta_w=2*pi/1000;
Ap=-(min(db(1:1:wp/delta_w+1)))
Ar=-round(max(db(wr/delta_w+1:1:501)))
```

运行后得切比雪夫1型数字滤波器

$$H(z)=\frac{0.0018(1+z^{-1})^4}{(1-1.4996z^{-1}+0.8482z^{-2})(1-1.5548z^{-1}+0.6493z^{-2})}$$

频率响应曲线如图2-52所示。

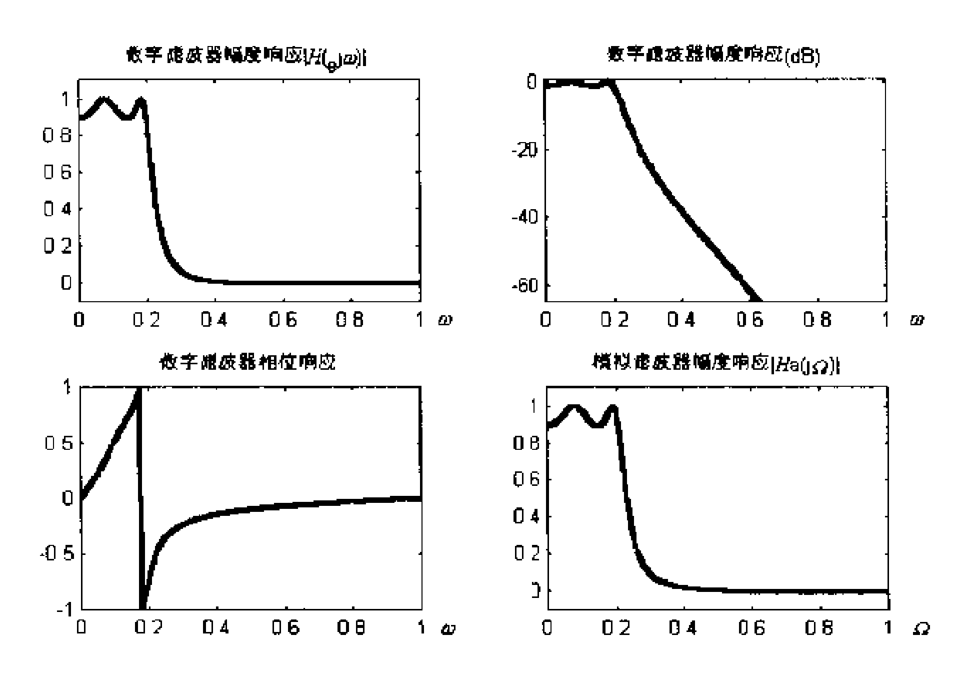


图2-52 双线性变换法设计的切比雪夫1型数字滤波器

运行后得切比雪夫2型数字滤波器。

$$H(z)=\frac{0.1797(1+0.5574z^{-1}+z^{-2})(1-1.0671z^{-1}+z^{-2})}{(1-0.4183z^{-1}+0.1503z^{-2})(1-1.1325z^{-1}+0.7183z^{-2})}$$

频率响应曲线如图2-53所示。

在实际应用中，经常用到数字低通、高通、带通、带阻滤波器，设计各种类型的滤波器可以有以下两种方法：①把一个模拟低通滤波器原型，经过频率变换法直接变换到所需的数字滤波器；②把一个模拟低通滤波器原型变换成数字低通滤波器，然后再用数字到数字的变换方法将它变换成所需的数字滤波器。

2.3.2.3 数字滤波器的原型变换法

由模拟低通滤波器直接变换成各种类型的数字滤波器的方法——原型变换法。模拟低通滤波器变换成数字低通滤波器，可以用前面的脉冲响应不变法和双线性变换法。

（1）模拟低通滤波器变换成数字高通滤波器

已知模拟低通滤波器 $H_a(s)$

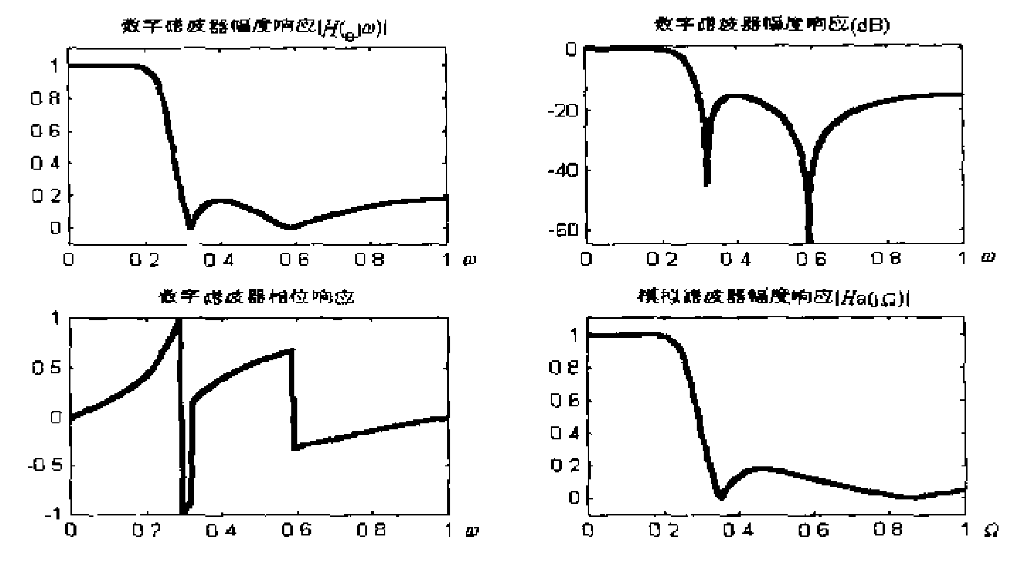


图 2-53 双线性变换法设计的切比雪夫 2 型数字滤波器

变换关系

$$s=\frac{2}{T}\cdot\frac{1+z^{-1}}{1-z^{-1}}$$

频率变换关系

$$\Omega=\frac{2}{T}\cot\frac{\omega}{2}$$

数字高通滤波器

$$H(z)=H_a(s)\Big|_{s=\frac{2}{T}\frac{1+z^{-1}}{1-z^{-1}}}$$

[例 7] 分别设计一个巴特沃斯和切比雪夫高通滤波器，要求通带截止频率为 0.6π，通带内衰减不大于 1dB，阻带起始频率为 0.4 π，阻带内衰减不小于 15dB。T=1。

解：预畸

$$\Omega_p=\frac{2}{T}\cot\frac{\omega_p}{2}=\frac{2}{T}\cot\left(\frac{0.6\pi}{2}\right)=\frac{2}{T}\cot(0.3\pi)$$

$$\Omega_r=\frac{2}{T}\cot\frac{\omega_r}{2}=\frac{2}{T}\cot\left(\frac{0.4\pi}{2}\right)=\frac{2}{T}\cot(0.2\pi)$$

模拟巴特沃斯滤波器

$$N\geqslant\frac{\lg[(10^{A_p/10}-1)/(10^{A_r/10}-1)]}{2\lg(\Omega_p/\Omega_r)}=3.7352$$

为满足指标，取阶数 N=4。

$$\Omega_c=\frac{\Omega_p}{\sqrt[2N]{(10^{A_p/10}-1)}}=1.7412\,\text{rad/s}$$

则

$$H_a(s)=\frac{8.7615}{(s^4+4.4958s^3+10.1060s^2+13.3074s+8.7615)}$$

用频率变换法求数字高通滤波器

$$H(z)=H_a(s)\Big|_{s=\frac{2}{T}\frac{1+z^{-1}}{1-z^{-1}}}=\frac{0.0751(1-2z^{-1}+z^{-2})^2}{(1+0.1562z^{-1}+0.4488z^{-2})(1+0.1124z^{-1}+0.0425z^{-2})}$$

上述方法可以用 buttord 函数确定巴特沃斯数字高通滤波器的阶次 N 和截止频率 ω_n，用 butter 函数把巴特沃斯模拟低通滤波器变换到巴特沃斯数字高通滤波器。

设计巴特沃斯型滤波器可采用下面几种形式：

```
[b,a]=butter(N,wn)%设计 N 阶低通数字滤波器;
[b,a]=butter(N,wn,'high')%设计 N 阶高通数字滤波器;
[b,a]=butter(N,wn,'bandpass')%设计 2N 阶带通数字滤波器;
[b,a]=butter(N,wn,'stop')%设计 2N 阶带阻数字滤波器。
```

设计切比雪夫 1 型滤波器可采用下面几种形式：

```
[b,a]=cheby1(N,Ap,wn)%设计 N 阶低通数字滤波器;
[b,a]=cheby1(N,Ap,wn,'high')%设计 N 阶高通数字滤波器;
[b,a]=cheby1(N,Ap,wn,'bandpass')%设计 2N 阶带通数字滤波器;
[b,a]=cheby1(N,Ap,wn,'stop')%设计 2N 阶带阻数字滤波器。
```

设计切比雪夫 2 型滤波器可采用下面几种形式：

```
[b,a]=cheby2(N,Ar,wn)%设计 N 阶低通数字滤波器;
[b,a]=cheby2(N,Ar,wn,'high')%设计 N 阶高通数字滤波器;
[b,a]=cheby2(N,Ar,wn,'bandpass')%设计 2N 阶带通数字滤波器;
[b,a]=cheby2(N,Ar,wn,'stop')%设计 2N 阶带阻数字滤波器。
```

wn 为 3dB 截止频率，其单位为π。

```
%s188.m
wp=0.6*pi;wr=0.4*pi;Ap=1;Ar=15;
[N,wn]=buttord(wp/pi,wr/pi,Ap,Ar)  %计算巴特沃斯滤波器阶次和截止频率
[b,a]=butter(N,wn,'high');%频率变换法设计巴特沃斯高通滤波器
[C,B,A]=dir2cas(b,a)  %数字高通滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(211);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(212);plot(w/pi,db);title('数字滤波器幅度响应(dB)');
```

运行后得到结果如图 2-54 所示。

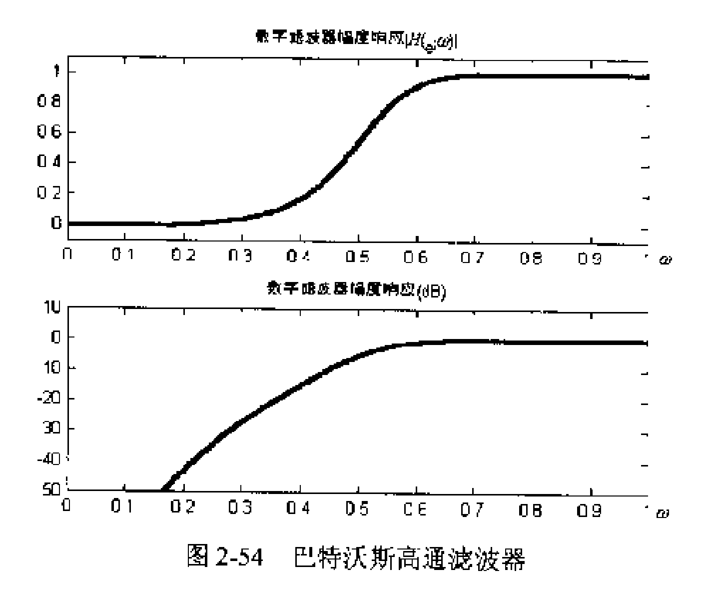


图 2-54 巴特沃斯高通滤波器

C =0.0751

B =1.0000 –2.0000 1.0000

```
    1.0000   -2.0000    1.0000
A = 1.0000    0.1562    0.4488
    1.0000    0.1124    0.0425
```

巴特沃斯数字高通滤波器为

$$H(z)=\frac{0.0751(1-2z^{-1}+z^{-2})^2}{(1+0.1562z^{-1}+0.4488z^{-2})(1+0.1124z^{-1}+0.0425z^{-2})}$$

用 cheb1ord 函数确定切比雪夫 1 型数字高通滤波器的阶次 N 和截止频率 ω_n，用 cheby1 函数把切比雪夫 1 型模拟低通滤波器变换到切比雪夫 1 型数字高通滤波器。

```
%s189.m
wp=0.6*pi;wr=0.4*pi;Ap=1;Ar=15;
[N,wn]=cheb1ord(wp/pi,wr/pi,Ap,Ar) %计算切比雪夫滤波器阶次和截止频率
[b,a]=cheby1(N,Ap,wn,'high');%频率变换法设计切比雪夫高通滤波器
[C,B,A]=dir2cas(b,a) %数字高通滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(211);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(212);plot(w/pi,db);title('数字滤波器幅度响应(dB)');
```

运行后得到结果如图 2-55 所示。

```
C = 0.0736
B = 1.0000   -2.0000    1.0000
    1.0000   -1.0000    0
A = 1.0000    0.5045    0.6188
    1.0000    0.4716    0
```

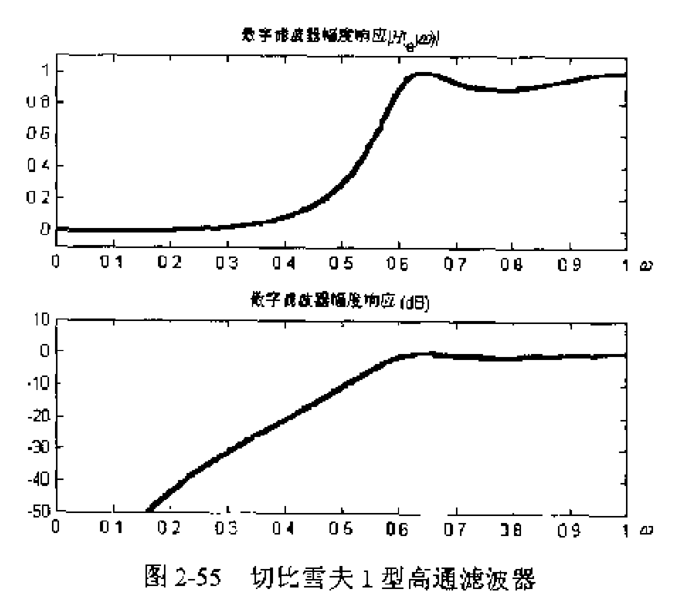


图 2-55　切比雪夫 1 型高通滤波器

切比雪夫 1 型数字高通滤波器为

$$H(z)=\frac{0.0736(1-2z^{-1}+z^{-2})(1-z^{-1})}{(1+0.5045z^{-1}+0.1688z^{-2})(1+0.4716z^{-1})}$$

（2）模拟低通滤波器变换成数字带通滤波器

已知模拟低通滤波器 $H_a(s)$

变换关系

$$s=\frac{1-2z^{-1}\cos\omega_0+z^{-2}}{1-z^{-2}}$$

频率变换关系

$$\Omega=\frac{\cos\omega_0-\cos\omega}{\sin\omega}$$

其中

$$\cos\omega_0=\frac{\cos[(\omega_2+\omega_1)/2]}{\cos[(\omega_2-\omega_1)/2]}$$

式中 ω_1——下截止频率；

ω_2——上截止频率。

数字带通滤波器为

$$H(z)=H_a(s)\Big|_{s=\frac{1-2z^{-1}\cos\omega_0+z^{-2}}{1-z^{-2}}}$$

［例 8］ 设计一个巴特沃斯带通滤波器，要求通带上下截止频率为 0.4π、0.3π，通带内衰减不大于 3dB，阻带上下起始频率为 0.5π、0.2π，阻带内衰减不小于 18dB。

解：根据通带指标确定中心频率

$$\cos\omega_0=\frac{\cos[(\omega_{p2}+\omega_{p1})/2]}{\cos[(\omega_{p2}-\omega_{p1})/2]}=\frac{\cos[(0.4\pi+0.3\pi)/2]}{\cos[(0.4\pi-0.3\pi)/2]}=0.4584$$

确定模拟滤波器截止频率

$$\Omega_c=\frac{\cos\omega_0-\cos\omega_{p2}}{\sin\omega_{p2}}=\frac{0.46-\cos(0.4\pi)}{\sin(0.4\pi)}=0.1548\text{rad/s}$$

根据上边带指标确定模拟巴特沃斯滤波器的阶次

$$\Omega_{p2}=\frac{\cos\omega_0-\cos\omega_{p2}}{\sin\omega_{p2}}=\frac{0.46-\cos(0.4\pi)}{\sin(0.4\pi)}=0.1548\ \text{rad/s}$$

$$\Omega_{r2}=\frac{\cos\omega_0-\cos\omega_{r2}}{\sin\omega_{r2}}=\frac{0.46-\cos(0.5\pi)}{\sin(0.5\pi)}=0.4584\ \text{rad/s}$$

$$N\geqslant\frac{\lg[(10^{A_{p2}/10}-1)/(10^{A_{r2}/10}-1)]}{2\lg(\Omega_{p2}/\Omega_{r2})}=1.96$$

为满足指标，取阶数 N=2。

模拟巴特沃斯滤波器为

$$H_a(s)=\frac{0.0251}{(s^2+0.2240s+0.0251)}$$

（3）频率变换法求数字带通滤波器

$$H(z)=H_a(s)\Big|_{s=\frac{1-2z^{-1}\cos\omega_0+z^{-2}}{1-z^{-2}}}=\frac{0.0201(1-z^{-2})^2}{1-1.6366z^{-1}+2.2373z^{-2}-1.3069z^{-3}+0.6413z^{-4}}$$

上述方法可以用 buttord 函数确定巴特沃斯数字带通滤波器的阶次 2N 和截止频率ω_n，此时ω_n具有两个元素。用 butter 函数把巴特沃斯模拟低通滤波器变换到巴特沃斯数字带通

滤波器。

```
%s190.m
wp=[0.3*pi,0.4*pi];wr=[0.2*pi,0.5*pi];Ap=3;Ar=18;
[N,wn]=buttord(wp/pi,wr/pi,Ap,Ar);
[b,a]=butter(N,wn,'bandpass');%频率变换法设计巴特沃斯滤波器
[C,B,A]=dir2cas(b,a) %数字带通滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(211);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(212);plot(w/pi,db);title('数字滤波器幅度响应(dB)');
```

运行后得到

```
b = 0.0213        0      -0.0426        0      0.0213
a = 1.0000   -1.6303      2.2183   -1.2919     0.6320
```

巴特沃斯带通滤波器为（直接型）

$$H(z)=\frac{0.0213(1-z^{-2})^{2}}{1-1.6303z^{-1}+2.218z^{-2}-1.2919z^{-3}+0.6320z^{-4}}$$

```
C =0.0213
B =1.0000   -2.0000    1.0000
   1.0000    2.0000    1.0000
A =1.0000   -0.6249    0.7840
   1.0000   -1.0053    0.8061
```

巴特沃斯带通滤波器为（级联型）

$$H(z)=\frac{0.0213(1-2z^{-1}+z^{-2})(1+2z^{-1}+z^{-2})}{(1-0.6249z^{-1}+0.7840z^{-2})(1-1.0053z^{-1}+0.8061z^{-2})}$$

频响曲线如图 2-56 所示，满足给定的技术指标。

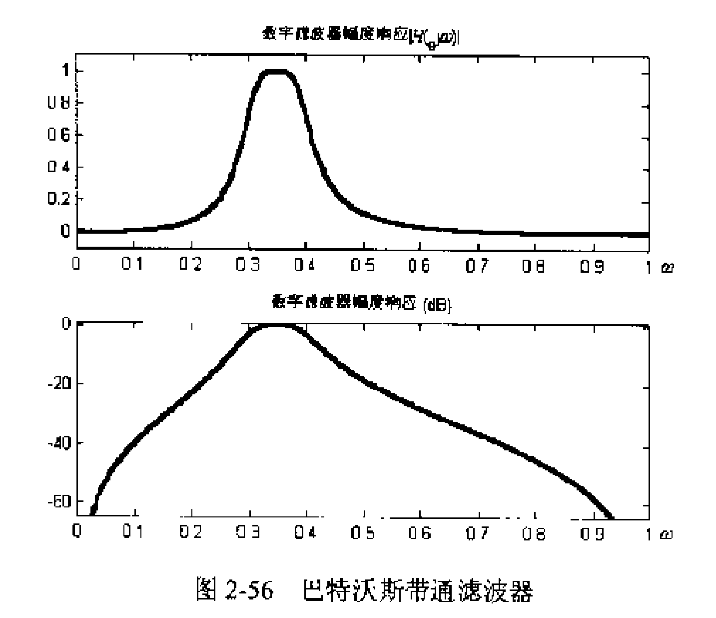


图 2-56　巴特沃斯带通滤波器

[例 9]　设计一个切比雪夫带通滤波器，要求通带上下截止频率为 0.4π、0.3π，通带内衰减不大于 3dB，阻带上下起始频率为 0.5π、0.2π，阻带内衰减不小于 18dB。

解：程序实现如下。

```
%s191.m
wp=[0.3*pi,0.4*pi];wr=[0.2*pi,0.5*pi];Ap=3;Ar=18;
[N,wn]=cheb1ord(wp/pi,wr/pi,Ap,Ar);
[b,a]=cheby1(N,Ap,wn,'bandpass');%频率变换法设计切比雪夫滤波器
[C,B,A]=dir2cas(b,a) %数字带通滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(211);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(212);plot(w/pi,db);title('数字滤波器幅度响应(dB)');
```

运行后得到结果如图 2-57 所示。

```
C =0.0112
B =1.0000   -2.0000    1.0000
   1.0000    2.0000    1.0000
A =1.0000   -0.6528    0.8985
   1.0000   -1.0728    0.9100
```

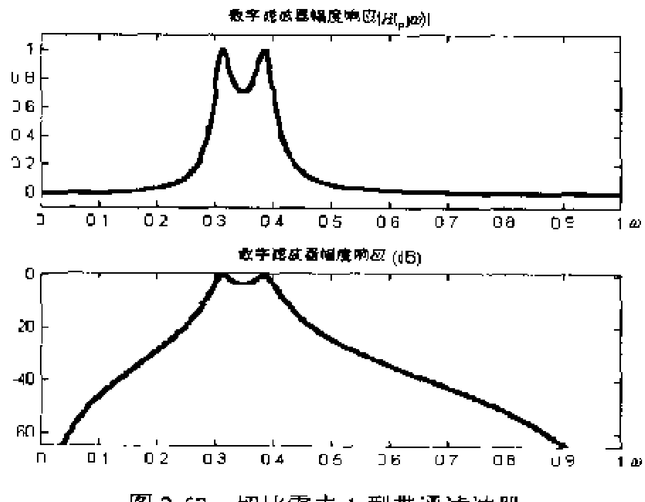


图 2-57 切比雪夫 1 型带通滤波器

切比雪夫 1 型带通滤波器为

$$H(z)=\frac{0.0112(1-2z^{-1}+z^{-2})(1+2z^{-1}+z^{-2})}{(1-0.6528z^{-1}+0.8985z^{-2})(1-1.0728z^{-1}+0.9100z^{-2})}$$

频响曲线如图 2-56 所示，满足给定的技术指标。

（4）模拟低通滤波器变换成数字带阻滤波器

已知模拟低通滤波器 $H_a(s)$

变换关系

$$s=\frac{1-z^{-2}}{1-2z^{-1}\cos\omega_0+z^{-2}}$$

频率变换关系

$$\Omega=\frac{\sin\omega}{\cos\omega-\cos\omega_0}$$

其中

$$\cos\omega_0=\frac{\cos[(\omega_2+\omega_1)/2]}{\cos[(\omega_2-\omega_1)/2]}$$

式中　ω_1——下截止频率；

ω_2——上截止频率。

数字带阻滤波器为

$$H(z)=H_a(s)\Big|_{s=\frac{1-z^{-2}}{1-2z^{-1}\cos\omega_0+z^{-2}}}$$

［例 10］　设计一个滤波器，要求通带上下截止频率为 0.8π、0.2π，通带内衰减不大于 1dB，阻带上下起始频率为 0.7π、0.4π，阻带内衰减不小于 30dB。

（1）巴特沃斯带阻滤波器；（2）切比雪夫 1 型带阻滤波器；（3）切比雪夫 2 型带阻滤波器。

```
%s192.m
wr=[0.4*pi,0.7*pi];wp=[0.2*pi,0.8*pi];Ap=1;Ar=30;
[N,wn]=buttord(wp/pi,wr/pi,Ap,Ar);
[b,a]=butter(N,wn,'stop');%频率变换法设计巴特沃斯带阻滤波器
[C,B,A]=dir2cas(b,a) %数字带阻滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(211);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(212);plot(w/pi,db);title('数字滤波器幅度响应 (dB)');
```

运行后得到结果如图 2-58 所示。

```
C =0.0394
B =1.0000    0.3554    1.0021
   1.0000    0.3544    0.9976
   1.0000    0.3521    1.0045
   1.0000    0.3501    0.9955
   1.0000    0.3478    1.0025
   1.0000    0.3468    0.9979
A =1.0000    1.3568    0.7928
   1.0000    1.0330    0.4633
   1.0000    0.6180    0.1775
   1.0000   -0.2493    0.1113
   1.0000   -0.6617    0.3755
   1.0000   -0.9782    0.7446
   1.0000   -1.4323    0.7778
```

图 2-58　巴特沃斯带阻滤波器

这是一个 12 阶巴特沃斯带阻滤波器。

```
%s193.m
wr=[0.4*pi,0.7*pi];wp=[0.2*pi,0.8*pi];Ap=1;Ar=30;
[N,wn]=cheb1ord(wp/pi,wr/pi,Ap,Ar);
[b,a]=cheby1(N,Ap,wn,'stop');%频率变换法设计切比雪夫1型带阻滤波器
[C,B,A]=dir2cas(b,a) %数字带阻滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(211);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(212);plot(w/pi,db);title('数字滤波器幅度响应 (dB)');
```

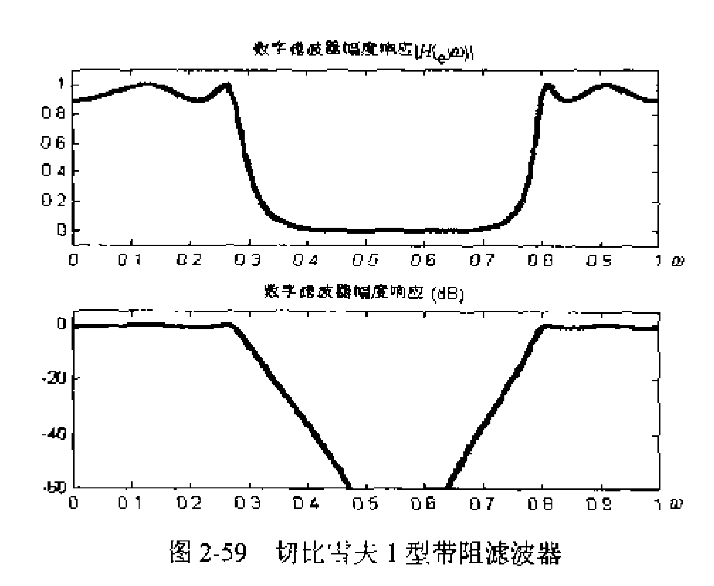


图 2-59 切比雪夫 1 型带阻滤波器

运行后得到结果如图 2-59 所示。

```
C =0.0463
B =1.0000    0.3514    1.0000
   1.0000    0.3512    1.0002
   1.0000    0.3511    0.9998
   1.0000    0.3509    1.0000
A =1.0000    1.5258    0.8846
   1.0000    1.5050    0.6292
   1.0000   -1.1997    0.8527
   1.0000   -1.2968    0.5242
```

这是一个 8 阶切比雪夫 1 型带阻滤波器。

```
%s194.m
wr=[0.4*pi,0.7*pi];wp=[0.2*pi,0.8*pi];Ap=1;Ar=30;
[N,wn]=cheb2ord(wp/pi,wr/pi,Ap,Ar);
[b,a]=cheby2(N,Ar,wn,'stop');%频率变换法设计切比雪夫 2 型带阻滤波器
[C,B,A]=dir2cas(b,a) %数字带阻滤波器级联型
[db,mag,pha,w]=freqz_m(b,a);%数字滤波器响应
subplot(211);plot(w/pi,mag);title('数字滤波器幅度响应|H(ej\omega)|');
subplot(212);plot(w/pi,db);title('数字滤波器幅度响应 (dB)');
```

运行后得到结果如图 2-60 所示。

```
C =0.1947
B =1.0000    1.2190    1.0000
   1.0000    0.7696    1.0000
   1.0000   -0.1017    1.0000
   1.0000   -0.6786    1.0000
A =1.0000    1.3301    0.7578
   1.0000    0.7348    0.3194
   1.0000   -0.3340    0.2511
   1.0000   -0.9544    0.7028
```

这是一个 8 阶切比雪夫 2 型带阻滤波器。

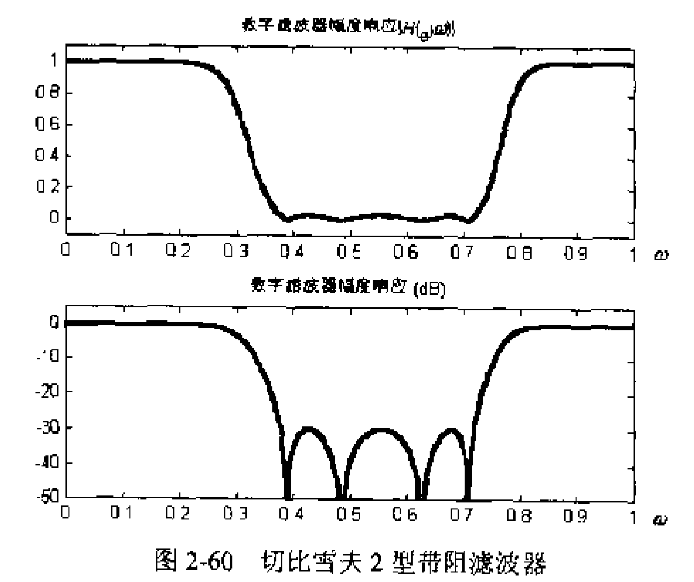


图 2-60 切比雪夫 2 型带阻滤波器

2.3.3 数字低通到数字滤波器的设计

直接由数字低通滤波器变换成各种类型的数字滤波器的方法——数字域频带变换。

给定数字低通滤波器系统函数 $H_{LP}(Z)$，通过一定的变换，设计其他各种不同类型的数字滤波器系统函数 $H(z)$，这种变换从 $H_{LP}(Z)$的 Z 平面到 $H(z)$ 的 z 平面的映射关系为

$$Z^{-1}=G(z^{-1})$$

假定 $H_{LP}(Z)$是稳定的和因果的滤波器，希望 $H(z)$也是稳定的和因果的滤波器。因此要求：

① $G(\cdot)$必须是 z^{-1} 的有理函数，这样 $H(z)$才是可实现的；

② Z 平面的单位圆必须映射到 z 平面的单位圆上；

③ Z 平面的单位圆内部必须映射到 z 平面的单位圆内部。

满足上述要求的有理函数为

$$Z^{-1}=G(z^{-1})=\pm\prod_{i=1}^{N}\frac{z^{-1}-\alpha_i^*}{1-\alpha_i z^{-1}}$$

式中 $G(z^{-1})$——全通函数；

N——全通函数的阶数；

α_i——全通函数的极点，可以是实数，也可以是共轭复数，但$|\alpha_i|<1$。

设θ和ω分别为 Z 平面与 z 平面的数字频率变量，当ω从 0 变到π时，全通函数相角变化量为 $N\pi$。选择合适的 N 和α_i，可以得到各种不同类型的变换。

2.3.3.1 数字低通到数字低通的变换

$H_{LP}(e^{j\theta})$和$H(e^{j\omega})$都是低通滤波器的频率响应，只不过截止频率不同。当θ 从 0 变到π时，相应的ω 也应从 0 变到π，按全通函数相角变化量为 $N\pi$，可确定 $N=1$，故

$$Z^{-1}=G(z^{-1})=\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$$

式中α 为实数。

频率关系

$$\omega=\arctan\left[\frac{(1-\alpha^2)\sin\theta}{2\alpha+(1+\alpha^2)\cos\theta}\right]$$

当$\alpha=0$时，$\omega=0$，频率变换是线性关系；当$\alpha<0$时，频率扩张，频率变换是非线性关系；当$\alpha>0$时，频率压缩，频率变换是非线性关系。

$$\alpha=\frac{\sin\left(\frac{\theta_c-\omega_c}{2}\right)}{\sin\left(\frac{\theta_c+\omega_c}{2}\right)}$$

式中 θ_c——低通原型的截止频率；

ω_c——变换后对应的截止频率。

变换后新的低通滤波器的系统函数为

$$H(z)=H_{LP}(Z)\Big|_{Z^{-1}=\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}}$$

2.3.3.2 数字低通到数字高通的变换

由低通到高通，只需将频率响应旋转 180°，全通函数的阶次 $N=1$。

变换后高通滤波器的系统函数为

$$H(z)=H_{LP}(Z)\Big|_{Z^{-1}=-\left(\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}\right)}$$

其中

$$\alpha=-\frac{\cos\left(\frac{\theta_c+\omega_c}{2}\right)}{\cos\left(\frac{\theta_c-\omega_c}{2}\right)}$$

2.3.3.3 数字低通到数字带通的变换

当低通数字频率θ从$-\pi$变到 0 时，相应的带通数字频率ω也应从 0 变到ω_0；当低通数字频率从 0 变到π时，相应的带通数字频率ω也应从ω_0变到π。因而ω由 0 变化到π，则θ必须变化 2π，全通函数的阶次 $N=2$。

变换后带通滤波器的系统函数为

$$H(z)=H_{\mathrm{LP}}(Z)\Big|_{Z^{-1}=-\left(\frac{z^{-2}-\frac{2\alpha k}{k+1}z^{-1}+\frac{k-1}{k+1}}{\frac{k-1}{k+1}z^{-2}-\frac{2\alpha k}{k+1}z^{-1}+1}\right)}$$

其中

$$\alpha=\frac{\cos\left(\frac{\omega_2+\omega_1}{2}\right)}{\cos\left(\frac{\omega_2-\omega_1}{2}\right)}=\cos\omega_0$$

$$k=\cot\left(\frac{\omega_2-\omega_1}{2}\right)\tan\frac{\theta_c}{2}$$

式中 ω_2 ——上截止频率；

ω_1 ——下截止频率；

ω_0 ——通带中心频率。

2.3.3.4 数字低通到数字带阻的变换

全通函数的阶次 $N=2$。

变换后带阻滤波器的系统函数为

$$H(z)=H_{\mathrm{LP}}(Z)\Big|_{Z^{-1}=-\left(\frac{z^{-2}-\frac{2\alpha}{1+k}z^{-1}+\frac{1-k}{1+k}}{\frac{1-k}{1+k}z^{-2}-\frac{2\alpha}{1+k}z^{-1}+1}\right)}$$

其中

$$\alpha=\frac{\cos\left(\frac{\omega_2+\omega_1}{2}\right)}{\cos\left(\frac{\omega_2-\omega_1}{2}\right)}=\cos\omega_0$$

$$k=\tan\left(\frac{\omega_2-\omega_1}{2}\right)\tan\frac{\theta_c}{2}$$

式中 ω_2 ——上截止频率；

ω_1 ——下截止频率；

ω_0 ——阻带中心频率。

［例11］ 用双线性变换法，设计了一个切比雪夫 1 型低通数字滤波器，满足要求在通带内频率低于 0.2π rad 时，允许幅度误差在 1dB 以内，在频率 0.3π rad 到π rad 之间的阻带衰减大于 15dB。它的系统函数为

$$H_{\mathrm{LP}}(Z)=\frac{0.0018(1+Z^{-1})^4}{(1-1.4996Z^{-1}+0.8482Z^{-2})(1-1.5548Z^{-1}+0.6493Z^{-2})}$$

现要求设计一个高通滤波器，容限与上述相同，但通带从 0.6π rad 开始。

解： 计算参数

$$\alpha=-\frac{\cos\left(\frac{\theta_c+\omega_c}{2}\right)}{\cos\left(\frac{\theta_c-\omega_c}{2}\right)}=-\frac{\cos\left(\frac{0.2\pi+0.6\pi}{2}\right)}{\cos\left(\frac{0.2\pi-0.6\pi}{2}\right)}=-0.3819$$

因此高通滤波器系统函数为

$$H(z)=H_{\mathrm{LP}}(Z)\Big|_{Z^{-1}=-\left(\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}\right)}=\frac{0.02426(1-z^{-1})^4}{(1+0.5561z^{-1}+0.7647z^{-2})(1+1.0416z^{-1}+0.4019z^{-2})}$$

为了从低通数字滤波器原型得到新数字滤波器的有理函数，必须实现有理代换，这个任务可以通过 zmapping 函数实现。

```
function[bz,az]=zmapping(bZ,aZ,Nz,Dz)
%从 Z 域到 z 域的频带变换
%实现:
% b(z)/a(z) = b(Z)/a(Z) |Z=N(z)/D(z)
bzord=(length(bZ)-1)*(length(Nz)-1);
azord=(length(aZ)-1)*(length(Dz)-1);
bz=zeros(1,bzord+1);
for k=0:bzord
   pln=[1];
   for l=0:k-1
      pln=conv(plr,Nz);
   end
   pld=[1];
   for l=0:bzord-k-1
      pld=conv(pld,Dz);
   end
   bz=bz+bZ(k+1)*conv(pln,pld);
end
az=zeros(1,azord+1);
for k=0:azord
   pln=[1];
   for l=0:k-1
      pln=conv(pln,Nz);
   end
   pld=[1];
   for l=0:azord-k-1
      pld=conv(pld,Dz);
   end
   az=az+aZ(k+1)*conv(pln,pld);
end
az1=az(1);az=az/az1;bz=bz/az1;
```

[例 12] 用 zmapping 函数实现例 1 中的高通滤波器。

```
%s195.m
wplp=0.2*pi;%低通数字通带频率
wrlp=0.3*pi;%低通数字阻带频率
Ap=1;%通带波动（dB）
Ar=15;%阻带波动（dB）
T=1;
Omegap=(2/T)*tan(wplp/2);%低通原型通带频率
Omegar=(2/T)*tan(wrlp/2);%低通原型阻带频率
[cs,ds]=afd_chb1(Omegap,Omegar,Ap,Ar);%模拟切比雪夫1型低通滤波器
[blp,alp]=bilinear(cs,ds,T);%双线性变换求低通数字滤波器
wphp=0.6*pi;%高通数字通带频率
alpha=-(cos((wplp+wphp)/2))/(cos((wplp-wphp)/2)) %低通到高通系数
Nz=-[alpha,1];Dz=[1,alpha];%变换关系
[bhp,ahp]=zmapping(blp,alp,Nz,Dz);%数字高通滤波器系统函数
[C,B,A]=dir2cas(bhp,ahp) %级联实现
[db,mag,pha,w]=freqz_m(blp,alp);%低通频率响应
subplot(2,2,1);plot(w/pi,mag);title('低通滤波器幅度响应');
subplot(2,2,2);plot(w/pi,db);title('低通滤波器幅度响应(dB)');
[db,mag,pha,w]=freqz_m(bhp,ahp);%高通频率响应
subplot(2,2,3);plot(w/pi,mag);title('高通滤波器幅度响应');
subplot(2,2,4);plot(w/pi,db);title('高通滤波器幅度响应(dB)');
```

运行后得到

```
alpha = -0.3820      C =0.0243
B =1.0000   -2.0000    1.0000
   1.0000   -2.0000    1.0000
A =1.0000    1.0416    0.4019
   1.0000    0.5561    0.7647
```

高通滤波器系统函数为

$$H(z)=\frac{0.0243(1-z^{-1})^4}{(1+0.5561z^{-1}+0.7647z^{-2})(1+1.0416z^{-1}+0.4019z^{-2})}$$

这个例子是根据数字低通滤波器的通带与阻带的边缘频率，设计出数字低通滤波器，再转换成数字高通滤波器。而高通滤波器的阻带边缘频率 $\omega_{\rm rhp}$ 满足方程

$$\alpha=\frac{\cos[(0.3\pi+\omega_{\rm rhp})/2]}{\cos[(0.3\pi-\omega_{\rm rhp})/2]}=-0.3820$$

它的解为 $\omega_{\rm rhp}=0.4586\pi$。

其频率响应曲线如图 2-61 所示。

在实际中，预期的高通滤波器的通带截止频率 $\omega_{\rm php}$ 与阻带截止频率 $\omega_{\rm rhp}$ 是已知的，需要找出低通原型的通带截止频率 $\omega_{\rm plp}$ 与阻带截止频率 $\omega_{\rm rlp}$。选择合理的通带频率 $\omega_{\rm plp}$、$\omega_{\rm php}$，确定 α，再根据 α 确定 $\omega_{\rm rlp}$

$$\omega_{\rm rlp}=\angle\left(-\frac{{\rm e}^{-{\rm j}\omega_{\rm rhp}}+\alpha}{1+\alpha{\rm e}^{-{\rm j}\omega_{\rm rhp}}}\right)$$

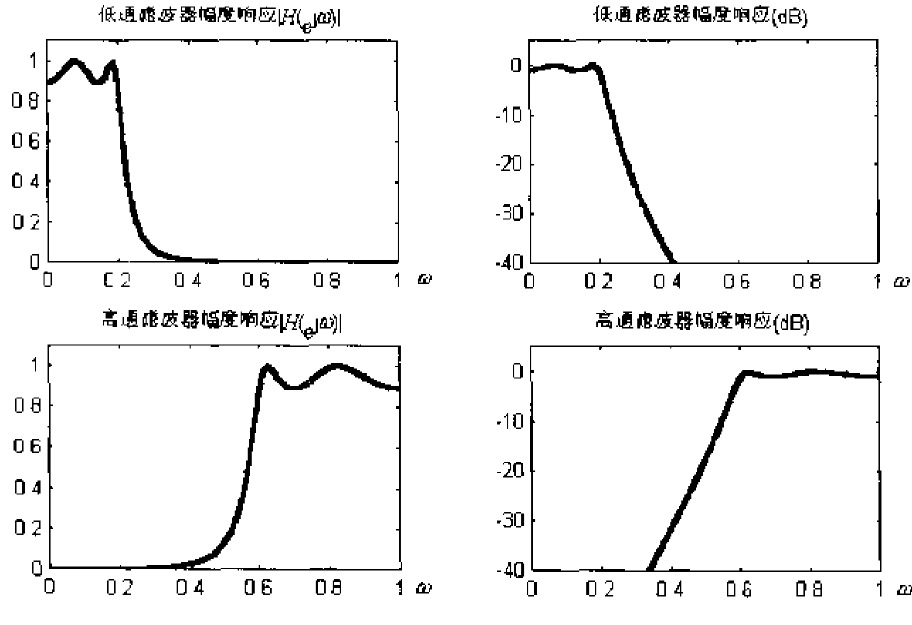


图 2-61　数字域变换法设计高通滤波器

设计低通滤波器，再用 zmapping 函数把它变换成高通滤波器。

[例 13]　用数字域频率变换法，设计一个切比雪夫 1 型高通数字滤波器，满足

$$\omega_{\mathrm{php}} = 0.6\pi \quad A_{\mathrm{p}} = 1\mathrm{dB}$$

$$\omega_{\mathrm{rhp}} = 0.4586\pi \quad A_{\mathrm{r}} = 15\mathrm{dB}$$

设低通滤波器的通带截止频率 $\omega_{\mathrm{plp}} = 0.2\pi$。

解：根据通带频率可得

$$\alpha = \frac{\cos[(0.2\pi + 0.6\pi)/2]}{\cos[(0.2\pi - 0.6\pi)/2]} = -0.3820$$

低通滤波器

$$\omega_{\mathrm{rlp}} = \angle\left(-\frac{\mathrm{e}^{-\mathrm{j}\omega_{\mathrm{rhp}}} + \alpha}{1 + \alpha\mathrm{e}^{-\mathrm{j}\omega_{\mathrm{rhp}}}}\right) = \angle\left(-\frac{\mathrm{e}^{-\mathrm{j}0.4586\pi} - 0.3820}{1 - 0.3820\alpha\,\mathrm{e}^{-\mathrm{j}0.4586\pi}}\right) = 0.3\pi$$

为了实现切比雪夫 1 型高通数字滤波器，上述过程可由 cheb1hpf 函数实现。

```
function[b,a]=cheb1hpf(wp,wr,Ap,Ar)
%用切比雪夫 1 型为原型，数字域频率变换法设计高通滤波器
wplp=0.2*pi;%低通数字通带频率
alpha=-cos((wplp+wp)/2)/cos((wplp-wp)/2);
wrlp=angle(-(exp(-j*wr)+alpha)/(1+alpha*exp(-j*wr)));%低通数字阻带频率
T=1;
Omegap=(2/T)*tan(wplp/2);%低通原型通带频率
Omegar=(2/T)*tan(wrlp/2);%低通原型阻带频率
[cs,ds]=afd_chb1(Omegap,Omegar,Ap,Ar);%模拟切比雪夫 1 型低通滤波器
[blp,alp]=bilinear(cs,ds,T);%双线性变换求低通数字滤波器
Nz=-[alpha,1];Dz=[1,alpha];
[b,a]=zmapping(blp,alp,Nz,Dz);

%s196.m
```

```
Ap=1; Ar=15;%通带、阻带波动（dB）
wp=0.6*pi;wr=0.4586*pi;%高通数字通带频率
[b,a]=cheb1hpf(wp,wr,Ap,Ar);
[C,B,A]=dir2cas(bhp,ahp)%级联实现
```

运行后得到

alpha = –0.3820

C =0.0243

B =1.0000　–2.0000　1.0000
　1.0000　–2.0000　1.0000

A =1.0000　1.0416　0.4019
　1.0000　0.5561　0.7647

高通滤波器系统函数为

$$H(z)=\frac{0.0243(1-z^{-1})^4}{(1+0.5561z^{-1}+0.7647z^{-2})(1+1.0416z^{-1}+0.4019z^{-2})}$$

结果与例 12 相同。

2.3.4 IIR 滤波器的应用实例

［例 14］ 用双线性变换法设计低通滤波器，满足技术指标：

$$\omega_p = 0.2\pi, A_p = 0.25\text{dB}$$
$$\omega_r = 0.4\pi, A_r = 50\text{dB}$$

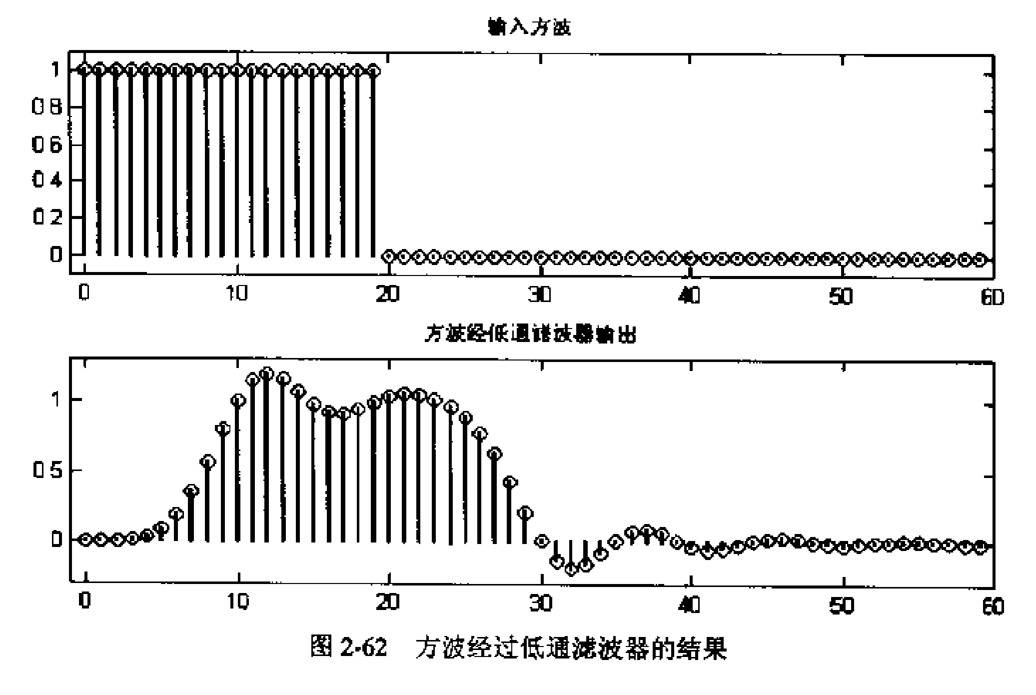


图 2-62 方波经过低通滤波器的结果

并对方波信号进行滤波。

```
%s197.m
wp=0.2*pi;wr=0.4*pi;Ap=0.25;Ar=50;
T=1;Omegap=(2/T)*tan(wp/2);Omegar=(2/T)*tan(wr/2);
[cs,ds]=afd_butt(Omegap,Omegar,Ap,Ar);%巴特沃斯原型
[b,a]=bilinear(cs,ds,T);%双线性变换法设计
```

```
[db,mag,pha,w]=freqz_m(b,a);
N=60;x=[ones(1,20),zeros(1,N-20)];
n=0:length(x)-1;
subplot(211);stem(n,x);title('输入方波');
y=filter(b,a,x);
subplot(212);stem(n,y);title('方波经低通滤波器输出');
```

运行后结果如图 2-62 所示。

[例 15] 一个含有高频噪声的信号保存在数据文件 noisbloc 中，试用脉冲响应不变法设计的低通滤波器对其进行除噪。

```
%s198.m
wp=0.2*pi;wr=0.4*pi;Ap=0.25;Ar=50;
T=1;Omegap=wp/T;Omegar=wr/T;
[cs,ds]=afd_chb1(Omegap,Omegar,Ap,Ar);%切比雪夫 1 型
[b,a]=imp_invr(cs,ds,T);%脉冲响应不变法
[db,mag,pha,w]=freqz_m(b,a);
load noisbloc;indx=1:1000;
x=noisbloc(indx);
subplot(211);plot(x);title('含噪声的输入信号');
y=filter(b,a,x);%直接型实现
subplot(212);plot(y);title('除噪后的输出信号');
```

运行结果见图 2-63。

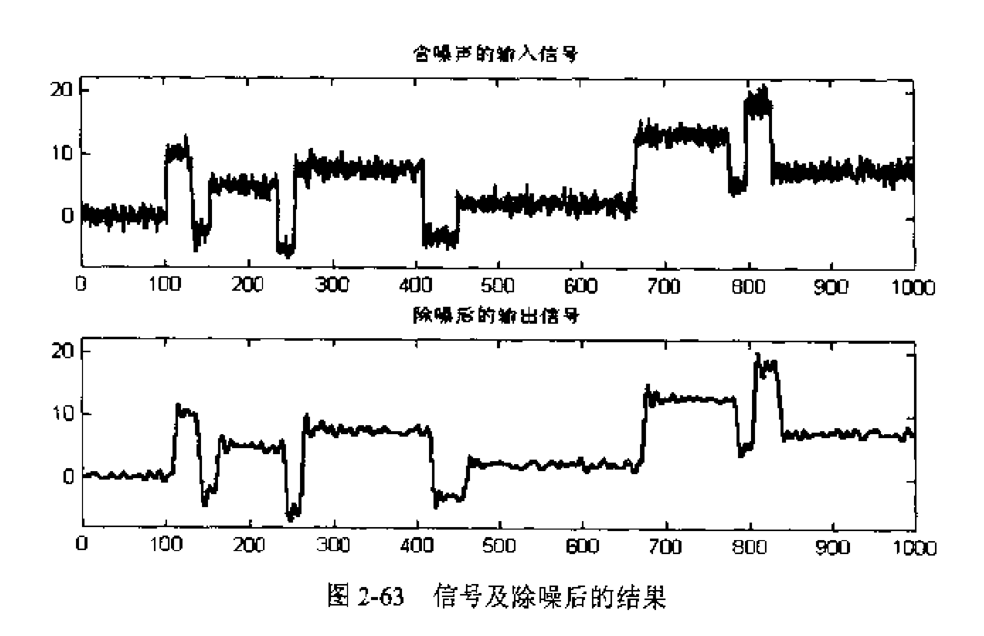


图 2-63 信号及除噪后的结果

[例 16] 模拟信号

$$x_a(t) = 5\sin(200\pi t) + 2\cos(300\pi t)$$

由系统 $x_a(t)$ → [A/D] → [$H(z)$] → [D/A] → $y_a(t)$ 处理，采样频率为 1000 Hz。

（1）设计一个最小阶数的 IIR 数字滤波器，以小于 1dB 的衰减通过 150Hz 的分量，以至少 40dB 抑制 100Hz 的分量。滤波器应有单调的通带和等波动的阻带，求出有理函数形式的系统函数，画出幅度响应（dB）。

（2）产生上述信号 $x_a(t)$的 150 个样本，通过上述所设计的滤波器得到输出序列，内插

此序列得到 $y_a(t)$。画出输入和输出信号的图并解释结果。

解：由题意用频率变换法设计切比雪夫 2 型数字高通滤波器

```
%s199.m
fp=150;fr=100;fs=1000;
wp=2*pi*fp/fs;wr=2*pi*fr/fs;
Ap=1;Ar=40;
[N,wn]=cheb2ord(wp/pi,wr/pi,Ap,Ar);
[b,a]=cheby2(N,Ar,wn,'high');
[C,B,A]=dir2cas(b,a)
[db,mag,pha,w]=freqz_m(b,a);
subplot(411);plot(w/pi,db);axis([0,1,-50,7]);
n=0:149;t=n/fs;
x=5*sin(2*pi*fr*t)+2*cos(2*pi*fp*t);
subplot(412);plot(t,x);axis([0,0.15,-7,7]);
y=filter(b,a,x);
subplot(413);stem(y);
ya=y*sinc(fs*(ones(length(n),1)*t-(n/fs)'*ones(1,length(t))));
subplot(414);plot(t,ya);axis([0,0.15,-2,2]);
```

运行后得到结果见图 2-64。

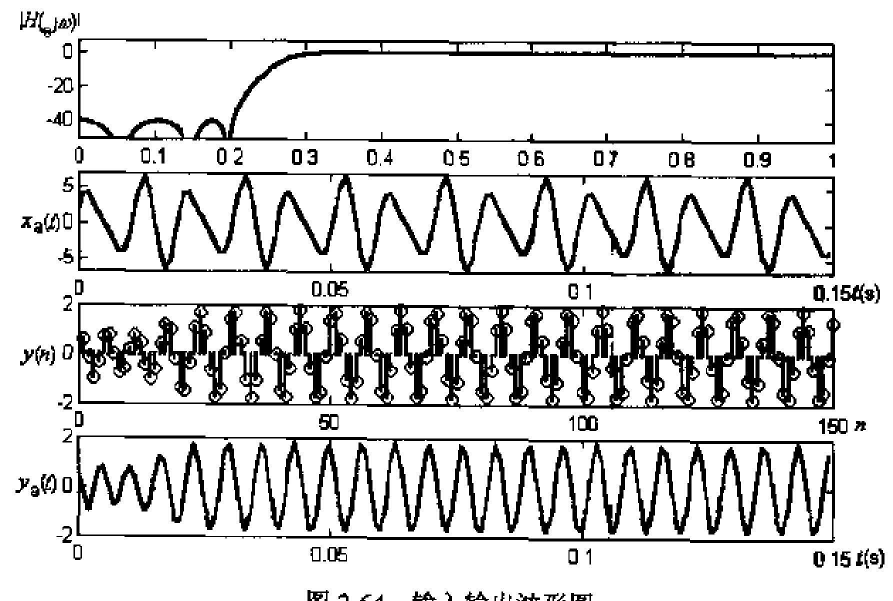


图 2-64　输入输出波形图

C =0.2942

B =	1.0000	−1.6284	1.0000
	1.0000	−1.7919	1.0000
	1.0000	−1.9708	1.0000
A =	1.0000	−1.0032	0.2706
	1.0000	−1.0213	0.4255
	1.0000	−1.1373	0.7517

系统函数

$$H(z)=\frac{0.2942(1-1.6284z^{-1}+z^{-2})(1-1.7919z^{-1}+z^{-2})(1-1.9708z^{-1}+z^{-2})}{(1-1.0032z^{-1}+0.2706z^{-2})(1-1.0213z^{-1}+0.4255z^{-2})(1-1.1373z^{-1}+0.7517z^{-2})}$$

从图 2-64 可见，输出信号抑制了输入信号的 100Hz 分量，基本上是 150Hz 的余弦信号，达到了信号处理的目的。

2.3.5 IIR 滤波器和 FIR 滤波器的比较

至今为止，我们讨论了 IIR 和 FIR 两种滤波器的设计方法，但在实际运应用时应该怎样去选择它们呢？下面对这两种滤波器作一比较。

从性能上说，IIR 滤波器系统函数的极点可以位于单位圆内的任何位置，因此可以用较低的阶数获得高的选择性，所用的存贮单元少，所以经济效率高。但是这个高效率是以相位的非线性为代价的。而 FIR 却可以得到严格的线性相位，然而 FIR 滤波器系统函数的极点固定在原点，所以只能用较高的阶数达到高的选择性。对于同样的滤波器设计指标，FIR 滤波器所需的阶数比 IIR 滤波器的阶数高 5～10 倍，结果成本较高，信号延时也较大。

从结构上看，IIR 滤波器必须采用递归结构，极点位置必须在单位圆内，否则系统将不稳定。在这种结构中，由于运算过程中对序列进行舍入处理，这种有限字长效应有时会产生寄生振荡。相反，FIR 滤波器采用非递归结构，不存在稳定性的问题，运算误差也较小，此外 FIR 滤波器可以采用快速傅里叶变换算法，在相同阶数的条件下，运算速度快得多。

从设计工具看，IIR 滤波器可以借助模拟滤波器的成果，因此，一般都有有效的封闭形式的设计公式可供准确计算，计算工作量比较小，对计算工具的要求不高。FIR 滤波器一般没有封闭形式的设计公式。窗口法仅仅对窗口函数可以给出计算公式，但计算通带阻带衰减无显式表达式。一般 FIR 滤波器的设计只有计算程序可循，因此对计算工具要求较高。

另外也应看到，IIR 滤波器虽然设计简单。但主要是用于设计具有片段常数特性的滤波器，如低、高、带通及带阻等，往往脱离不了模拟滤波器的格局。而 FIR 滤波器则要灵活得多，尤其它能易于适应某些特殊的应用，如构成微分器或积分器，或用于巴特沃斯、切比雪夫等逼近不可能达到预定指标的情况。例如，由于某些原因要求三角形振幅响应或一些更复杂的幅频响应，因而有更大的适应性和更广阔的天地。

从上面的简单比较我们可以看到，IIR 和 FIR 滤波器各有所长，所以在实际应用时应该从多方面考虑来加以选择。例如，从使用要求上来看，在对相位要求不敏感的场合，如语言通讯等，选用 IIR 较为合适，这样可以充分发挥其经济高效的特点，而对于图像信号处理、数据传输等以波形携带信息的系统，则对线性相位要较高，采用 FIR 滤波器较好。当然，在实际应用中应考虑经济上的要求。

第3章　数字信号处理的应用及实现

3.1　一维信号处理的应用

3.1.1　数据的采集

数字信号处理的对象是在采样时钟的控制之下，通过 A/D 转换器以一定的采样率对模拟信号进行采样得到的。根据采样定理，采样频率必须大于模拟信号最高频率的 2 倍。但在许多情况下，要求信号以不同的频率采样，改变取样率虽然可行，但并不可取。这时需要对采样数据进行处理：用抽取的方法降低其采样率（下采样）；用内插的方法提高其采样率（上采样）。

3.1.1.1　重采样

重采样包括上采样和下采样，改变采样率通常是为了达到以下目的。

① 降低数字信号处理的计算量，提高处理速度。提高处理速度是实现实时处理的首要目标之一。在信号处理过程中，常会出现插据的冗余，这时如果不去掉冗余的数据，就会影响处理的速度，例如数据系列经低通数字插波后，其最高频率就比原来的最高频率有所降低，这时就会产生或增加冗余数据，成为过采样，此时可通过调整采样率，使它等于或稍高于新的奈奎插特频率的 2 倍，这样所要处理的数据就会显著地下降。在语音信号处理中，使用下采样来减少语音数插，可用作快速的语音参插估计或减少数据传输量；在需要时，又可以通过上采样方法用这些经下采样的插据重建原始语音信号。

② 使拥有不同采样率的系统互相兼容。例如激光唱盘的采样率为 44.1kHz，如果要将它在一个 48kHz 的立体声音响系统里播放，就必须使它的采样率上升为 48kHz。

（1）下采样

设插字信号为$x(n)=x(t)|_{t=nT}$，如果希望将采样频率f_s减小 M 倍，那么最简单的方法是每 M 个点中抽取一个，依次组成一个新的序列$x'(n)$。即

$$x'(m)=\begin{cases}x(Mm) & ,m=0,1,\cdots,N' \\ 0 & ，其他\end{cases}$$

其中 N'为小于 N/M 的最大数数。

经下采样后数据的采样率为f_s/M，如果原始输入信号的最高频率大于$f_s/(2M)$，则下采样时会产生频率混叠。为了避免混叠，在进行下采样之前需对数据进行抗混叠滤波，使其最高频率分量低于下采样后的奈来斯特频率$f_s/(2M)$。

用 MATLAB 实现下采样的函数为 decimate，将式为：

`Y=decimate(X,R)`	返回的是向量 X 的重采样序列，采样率为原来的 $1/R$ 倍，因此 X 的长度为 Y 的 R 倍。
`Y=decimate(X,R)`	使用的滤波器是一 8 阶切比雪夫 1 型低通滤波器，其截止频率为 $0.8*(F_s/2)/R$。

```
Y=decimate(X,R,N)          使用的是 N 阶切比雪夫 1 型低通滤波器
Y=decimate(X,R,'FIR')      使用的是 30 点的 FIR 滤波器，这一滤波器是由函数 fir1(30,1/R)
                           产生的。
Y=decimate(X,R,N,'FIR')    使用的是 N 点 FIR 滤波器。
```

[例 1] 有一信号的采样为 sig，其最高频率为 150Hz，采样频率为 10000Hz，采样点数为 8000，试将其采样频率降为 1000Hz。

解：将采样频率降低为原来的 1/10，要用到的滤波器的截止频率很低，为了达到这一要求，滤波器的阶数必然很大，计算量也将很大，因此可采用分两级下采样的方法。

```
%s301.m
T=10000;n=0:8000;
sig=sin(2*pi*75*n/T)+sin(2*pi*150*n/T);
sig1=decimate(sig,5);
sig2=decimate(sig1,2);
subplot(311);stem(sig(1:200));
subplot(312);stem(sig1(1:40));axis([0,40,-2,2])
subplot(313);stem(sig2(1:20));axis([0,20,-2,2])
```

运行结果如图 3-1 所示，实现了对信号的 10 倍抽取。

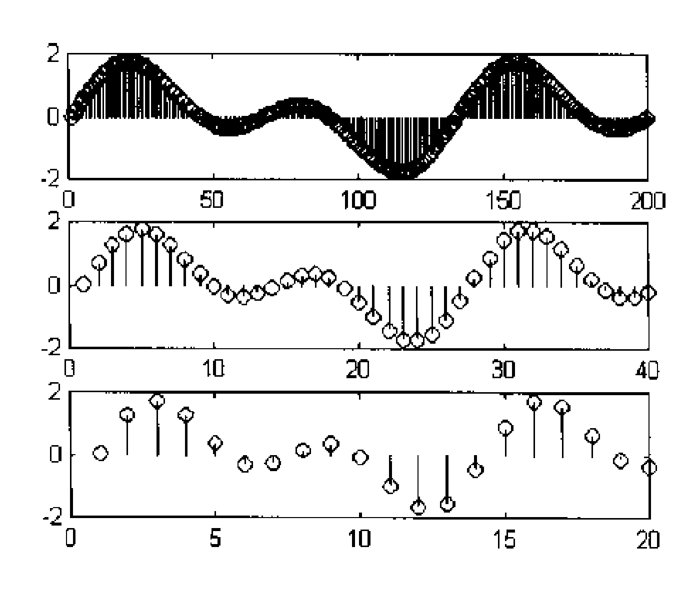


图 3-1 信号的整数倍抽取

（2）上采样

如果将数字信号 $x(n)$ 的采样频率 f_s 增加 L 倍，得到 $v(n)$，$v(n)$ 即是对 $x(n)$ 的插值。插值最简单的方法是在 $x(n)$ 每相邻两个点之间补 L-1 个零，然后再进行低通滤波处理。即

$$v(m)=\begin{cases}x(m/L) & ,m=0,\pm L,\pm 2L\cdots \\ 0 & ,\ \text{其他}\end{cases}$$

信号的插值虽然是零，但经过低通滤波后，信号得到了平滑，零值可能不再是零。

用 MATLAB 实现上采样的函数为 interp，格式如下。

```
Y=interp(X,R)  返回的是以原来 R（应为整数）倍的采样率重采样的序列。重采样得到的序列 Y
               的长度是 X 的 R 倍。
```

```
%s302.m
```

```
T=1000;n=0:8000;
sig=sin(2*pi*75*n/T)+sin(2*pi*150*n/T);
sig1=interp(sig,5);
subplot(211);stem(sig(1:20));
subplot(212);stem(sig1(1:100));axis([0,100,-2,2])
```

运行结果如图 3-2 所示，实现了对信号的 5 倍插值。

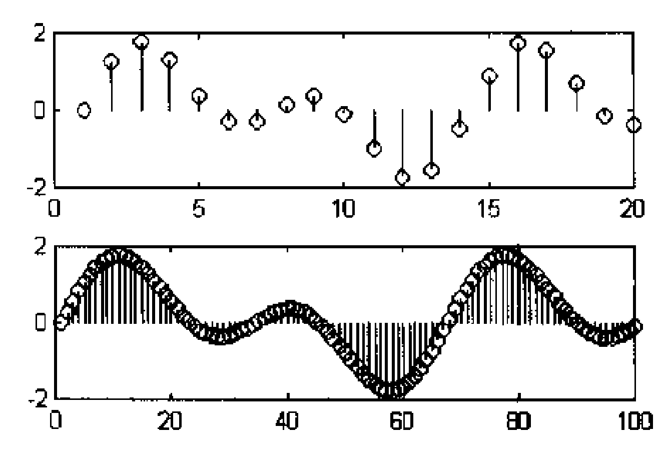


图 3-2 信号的整数倍插值

3.1.1.2 采样率的非整数倍转换

对于要将采样率做非整数倍改变的情况，可以将上采样和下采样结合起来实现，如先对 $x(n)$ 做 M 倍的抽取，再做 L 倍的插值来实现，或是先做 L 倍的插值，再做 M 倍的抽取。一般来说，抽取使数据点减少，会产生信息的丢失，因此合理的方法是先对信号做插值，然后再抽取，如图 3-3(a)所示。

这时插值后的去镜值滤波器和抽取前的抗混叠滤波器工作在级联状态，且所处理的信号的采样率都是 Lf_s，因此可将他们合二为一，如图 3-3(b)所示。

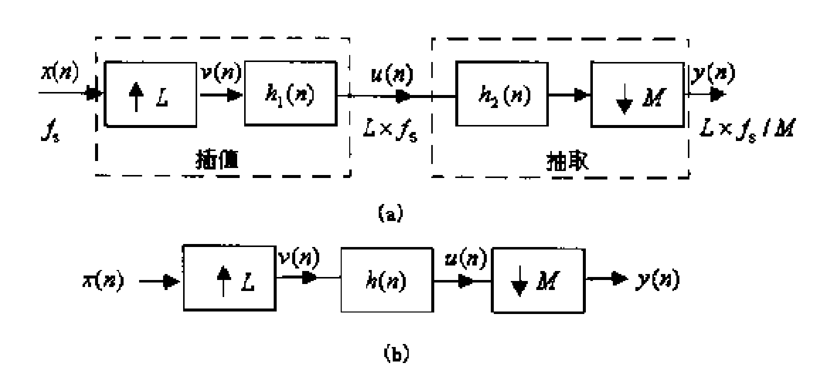


图 3-3 插值和抽取的级联实现

用 MATLAB 改变信号采样率的函数为 resample，格式如下。

`Y=resample(X,P,Q)` 返回的是向量 X 经重采样后的结果，重采样后的采样率为原来的 P/Q 倍。返回向量的长度是向量 X 的 P/Q 倍，P 和 Q 必须是正整数。

在重采样时 resample 使用一个抗混叠低通 FIR 滤波器对 X 进行滤波，滤波器由 FIR1 设计，使用的窗是凯塞窗。

Y=resample(X,P,Q,N)　在重采样时，x(n)的两边都取N点。Resample 使用的滤波器的长度与N成正比，通过增大N可获得更好的精度，但是是以牺牲计算时间为代价。N的缺省值为10。

```
%s303.m
T=1000;n=0:8000;
sig=sin(2*pi*75*n/T)+sin(2*pi*150*n/T);
sig1=resample(sig,7,2);
subplot(211);stem(sig(1:20));
subplot(212);stem(sig1(1:70));axis([0,70,-2,2])
```

运行结果如图3-4所示，实现了对信号的3.5倍重采样。

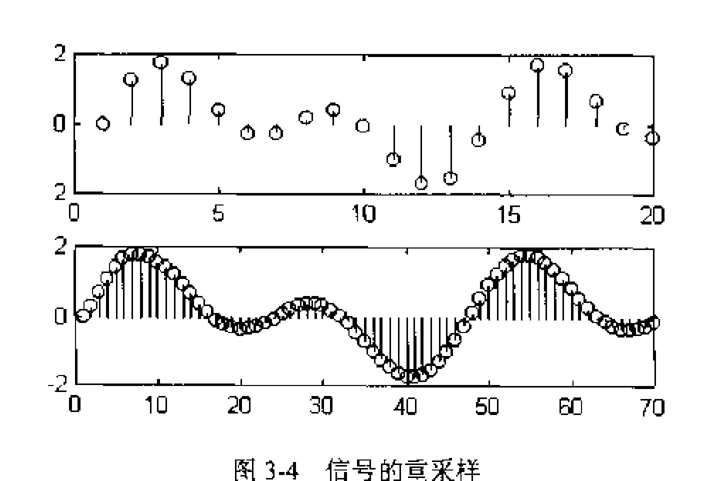


图3-4　信号的重采样

3.1.2　信号消噪处理

一个含有噪声的一维信号的模型可以表示为如下形式

$$s(i)=f(i)+\sigma e(i),i=0,\cdots,n-1$$

式中　$f(i)$——真实信号；

$e(i)$——噪声；

$s(i)$——含噪声的信号。

3.1.2.1　对平稳信号消噪

在实际工程应用中，有用信号通常表现为低频信号或是一些比较平稳的信号，而噪声信号则通常表现为高频信号，所以可以用加低通滤波器的方法进行滤波。

[例2]　工厂为了考察用电情况，对某三天的电网电压值进行监测，得到一个电压采样序列，但由于在采样过程中，监测设备出现了一些故障，使得所采集的信号引入了噪声，用傅里叶变换将这种由于仪器故障引起的噪声进行消噪处理。

针对本问题，选用不同截止频率的低通滤波器进行消噪处理，程序如下。

```
%s304.m
load leleccum;
s=leleccum(1:3920);
ls=length(s);
subplot(221);plot(s);
```

```
Xs=fft(s); %对含噪声信号进行傅里叶变换
absXs=abs(Xs);
subplot(222);plot(absXs);
indd2=50:3870; %进行低通滤波，滤波频率为 0~50 的相对频率
Xs(indd2)=zeros(size(indd2));
xd1=ifft(Xs);xd1=real(xd1);
subplot(223);plot(xd1);
indd2=20:3900; %进行低通滤波，滤波频率为 0~50 的相对频率
Xs(indd2)=zeros(size(indd2));
xd2=ifft(Xs);
xd2=real(xd2);
subplot(224);plot(xd2);
```

结果如图 3-5 所示，从中可以看出：低通滤波截止频率低时信号较光滑，它可能失去信号中的有用成分。

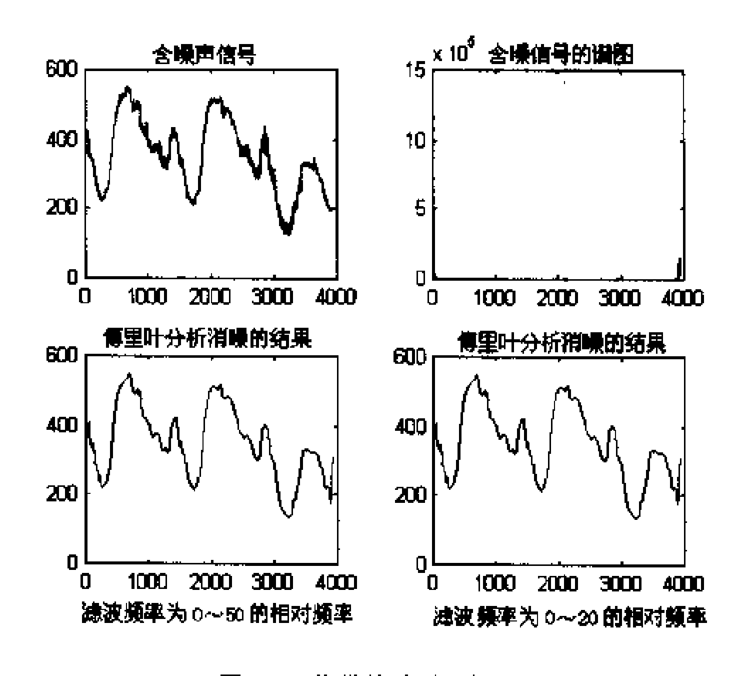


图 3-5 信号的消噪（例 2）

3.1.2.2 对非平稳信号消噪

在实际工程应用中，所分析的有用信号可能包含许多尖峰和突变部分，并且噪声也不是平稳的白噪声。对这种信号的分析，首先需要作信号的预处理，将信号的噪声部分去除，提取有用信号。

[例 3] 一含有噪声的矩形波信号，用傅里叶变换将噪声进行消噪处理。参考程序如下。

```
%s305.m
snr=5;%设置信噪比
init=2055615866;%设置随机数初值
[x,s]=wnoise(1,11,snr,init);%产生原始信号和含白噪声信号
s=s(1:1000);x=x(1:1000);
subplot(321);plot(x);axis([0,1000,-10,25]);title('原始信号');
```

```
subplot(322);plot(s);axis([0,1000,-10,25]);title('含噪声信号');
Xx=fft(x);%对原信号进行傅里叶变换
absXx=abs(Xx);
Xs=fft(s);%对含噪声信号进行傅里叶变换
absXs=abs(Xs);
subplot(323);plot(absXx);title('原始信号的谱图');
subplot(324);plot(absXs);title('含噪信号的谱图');
indd2=100:900;%进行低通滤波
Xs(indd2)=zeros(size(indd2));
xd=ifft(Xs);xd1=real(xd);
subplot(325);plot(xd1);axis([0,1000,-10,25]);
title('用傅里叶分析消噪的结果');
```

从图 3-6 可以看出，用傅里叶分析的方法进行非平稳信号消噪，它不能将信号的高频部分和由噪声引起的高频干扰加以区分，将一部分有用信号当作噪声滤掉了，但可以保存有用信号的突变部分。

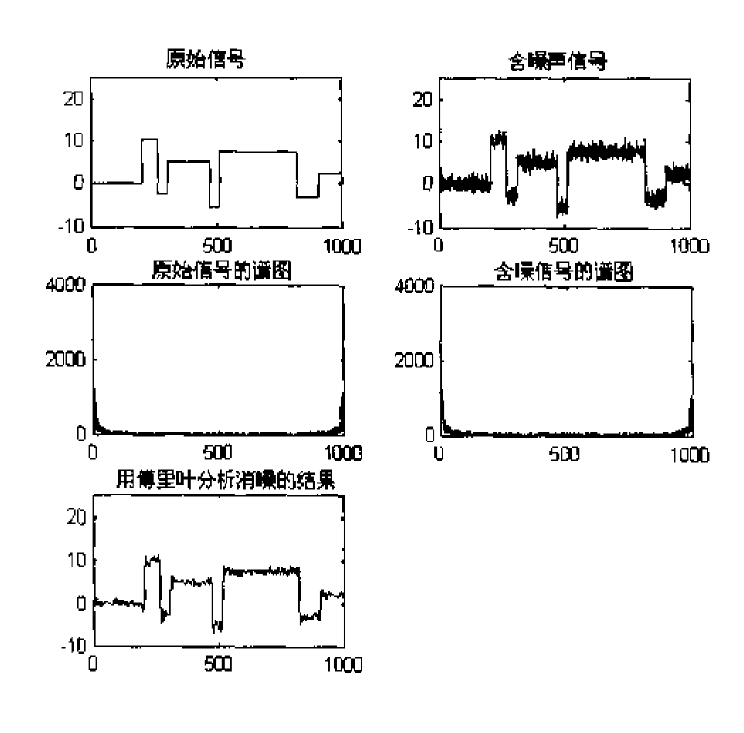


图 3-6　信号的消噪（例 3）

3.1.3　信号特定频率的提取

一个信号是由几个不同频率的正弦信号所组成，用傅里叶分析可以有效地分辨出这些不同频率的信号。程序如下。

```
%s306.m
N=200;
x=0:2*pi/N:2*pi-2*pi/N;
s=sin(5*x)+sin(20*x)+sin(40*x);
subplot(311);plot(s);
```

```
Y=fft(s);m=abs(Y);
subplot(312);plot(m);
M=15;L=10;L1=N-2*(M+L);%加带通滤波器
H=[zeros(1,M),ones(1,L),zeros(1,L1),ones(1,L),zeros(1,M)];
S=Y.*H;
s1=ifft(S);
s1=real(s1);
subplot(313);plot(s1);axis([0,200,-1.1,1.1]);
```

结果如图 3-7 所示，从中可见：信号含有三种频率成分，用带通滤波器提取其中的一种频率。

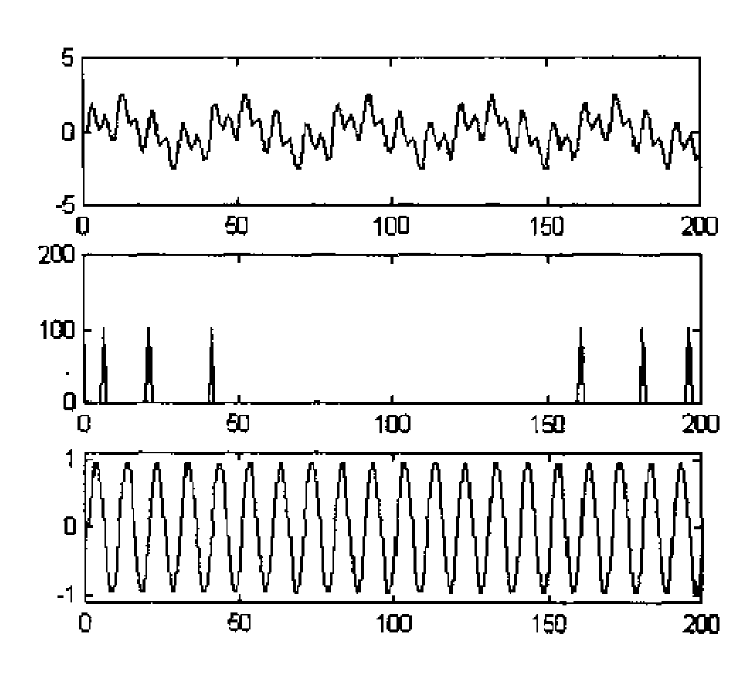


图 3-7 提取特定频率

[例4] 一个采集的实际信号（first.dat），从其第 9000 个数据开始取 500 个数据。（1）提取相对频率为 0～25 的频率成分；（2）提取相对频率为 50～150 的频率成分。

```
%s307.m
fid=fopen('first.dat','r');
st=fseek(fid,9000,'bof');
s=fread(fid,500,'int16');
N=length(s);n=0:N-1;
subplot(221);plot(n,s);
X1=fft(s);
m=abs(X1);
subplot(222);plot(n,m);
H=[ones(1,50),zeros(1,400),ones(1,50)]; %低通滤波器
Y=H'.*X1;s1=ifft(Y);s2=real(s1);
subplot(223);plot(n,s2);
H=[zeros(1,100),ones(1,100),zeros(1,100),ones(1,100),zeros(1,100)]; %带通滤波器
Y=H'.*X1;s1=ifft(Y);s2=real(s1);
```

```
subplot(224);plot(n,s2);
```

提取结果如图 3-8 所示。

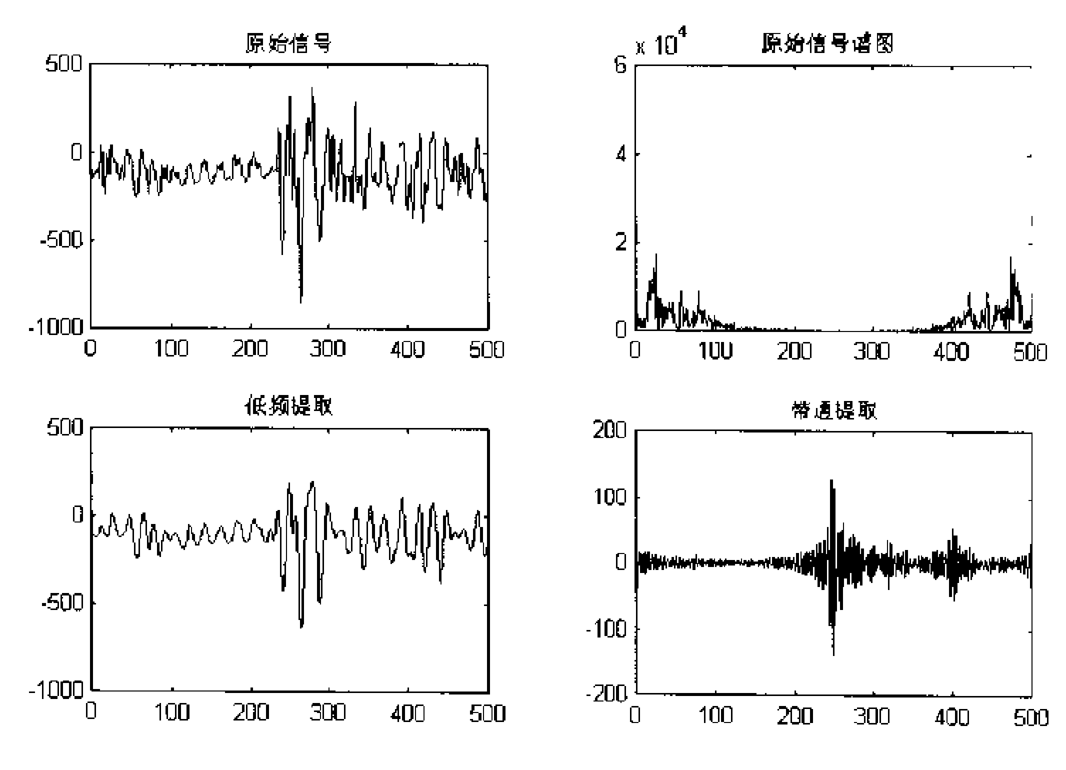


图 3-8 信号特定频率的提取

3.1.4 信号特定频率区间的抑制

滤波器的主要作用是按设计者的目的，突出或抑制一些频率。

[例 5] 人体心电图在测量过程中往往受到工业高频的干扰，所以必须经过低通滤波器处理后，才能判断心脏功能的有用信息。下面是一实际心电图信号的采样序列：$x(n)$ ={–4, –2,0,–4,–6,–4,–2,–4,–6,–6, –4,–4,–6,–6,–2,6,12,8,0,–16,–38,–60,–84,–90,–66,–32,–4,–2,–4,8,12,12, 10,6,6,4,0,0,0,0,0,–2,–2,0,0,–2,–2,–2,–2,0}，其中存在高频干扰。用低通滤波器滤除其中的干扰成分。

程序如下。

```
%s308.m
x=[-4,-2,0,-4,-6,-4,-2,-4,-6,-6,-4,-4,-6,-6,...
   -2,6,12,8,0,-16,-38,-60,-84,-90,-66,-32,-4,-2,...
   -4,8,12,12,10,6,6,4,0,0,0,0,0,-2,-2,0,0,-2,-2,-2,-2,0];
subplot(221);plot(x);axis([0,50,-100,20]);
fp=40;fr=50;Ap=0.5;Ar=40;fs=200;
[N,Wn]=buttord(fp/(fs/2),fr/(fs/2),Ap,Ar);
[b,a]=butter(N,Wn);
y=filter(b,a,x);
subplot(222);plot(y);axis([0,50,-100,20]);
subplot(223);psd(x,[],200);
subplot(224);psd(y,[],200);
```

分析这一段程序可知包括以下几部分：

① 绘制原信号的图形；

② 用设计的滤波器对原信号滤波，绘制滤波后的图形；

③ 绘制原信号功率谱图形；

④ 绘制滤波以后信号的功率谱图形。

结果如图3-9所示。

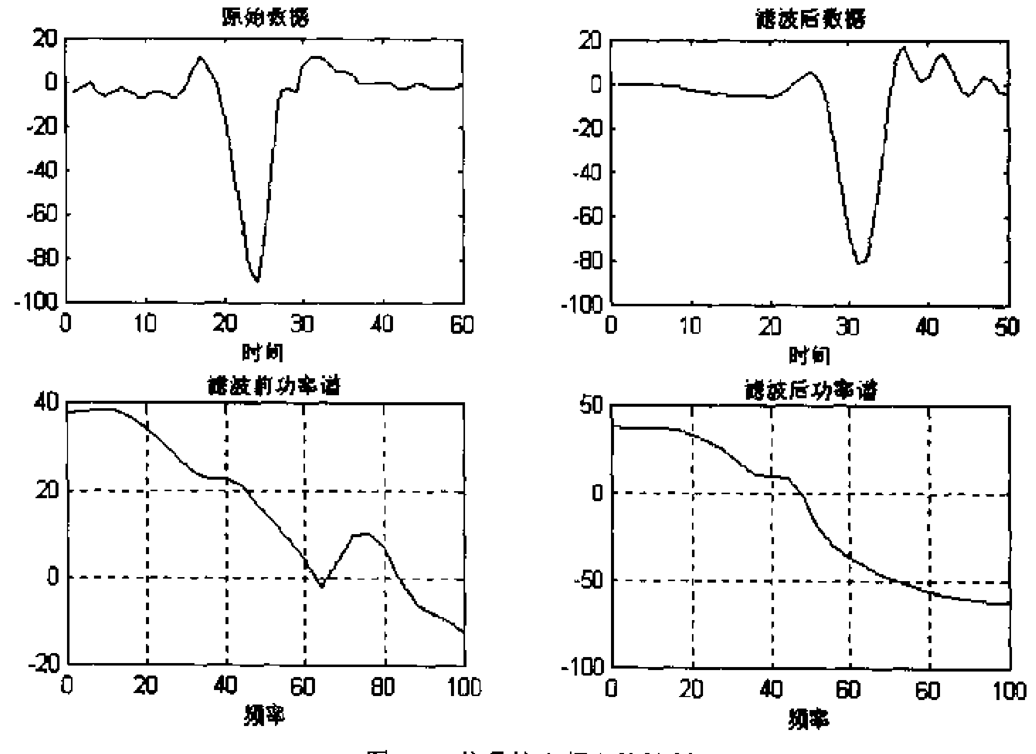


图3-9 信号特定频率的抑制

3.1.5 数字音频回响处理器的实现

在音乐厅内听到的声音由三部分组成：从声源直接传来的声音，经过少次墙壁反射传来的前期声音，经过多次墙壁反射传来的后期声音。这三种声音信号到达的先后次序不同，并互相混叠。如果把整个音乐厅当作一个对声源发出的声音进行处理的滤波器，它的单位脉冲响应反映了它的回响性能的好坏。

脉字回响处理器可以看做由一些基本模块组成。这些模块包括延时相加模块，该模块模拟声音经墙脉反射，产生一定的延时，并与直接到达的声音相加。该模块具有如下的脉冲响应。

$$y(n)=x(n)+ay(n-D)$$

其中a表示经过反射的衰减，D为反射造成的脉时。对应的系统函数为

$$H(z)=\frac{1}{1-az^{-D}}$$

该系统对应的框图如图3-10所示。

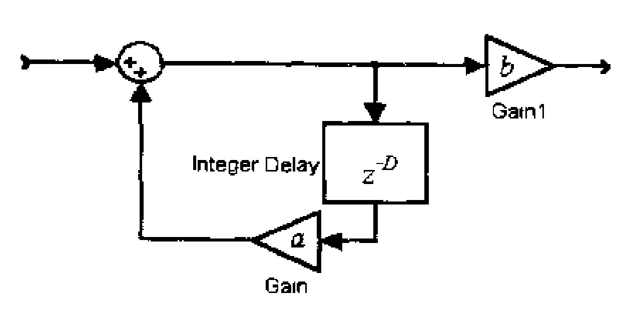


图3-10 第一种延迟相加模块

这种模块的频谱特性在一些频率点呈尖峰值，使它对声音信号中信对应频率加

强，而其他频率成分相应弱化。为消除这种情况，引入另外一种模块。

$$y(n)=ay(n-d)-ax(n)+x(n-D)$$

对应的系统函数为

$$H(z)=\frac{-a+z^{-D}}{1-az^{-D}}$$

系统对应的框图如图 3-11 所示。

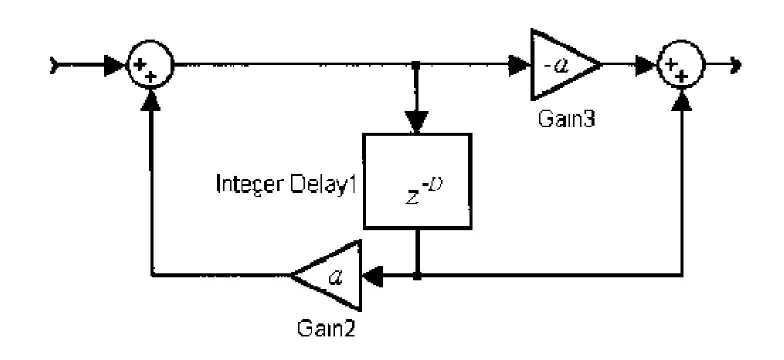


图 3-11 第二种延迟相加模块

通过适当的组合这两个基本模块，可以获得对实际音乐厅回响性能较好的数字滤波器。由于本例实现的数字滤波器中，要重复用到上面的两个基本模块。因此，可以利用 Simulink 的子系统创建功能，把这两个模块封装成子系统，以便在具体实现时进行调用。

封装第一个延迟相加模块。首先，按照上面所示的框图用 Simulink 创建好模块，该子模块有三个参数：增益 a、b 和延时量 D，分别双击对应的单元，设置好单元的参数。其次，选中该子模块所有的单元。点击菜单命令"Edit|Create Subsystem"，创建子系统。然后，将创建好的子系统加以保存，并点击菜单命令"Edit|Mask Subsystem"，封装子系统，并设置该子模块的参数；这时会弹出一个属性页，通过它来设置子模块的属性。将属性页切换到"Initialization"页上，通过该页来添加模块的参数。我们以添加参数"a"为例进行说明。点击"Add"按钮，表示添加参数；在"Prompt"编辑框内，填入该参数的名称；在"Variable"框内，填入该参数对应的变量名，在"Control type"框中，选取设置参数值时采用的控件；在"Assignment"框中，选取取值的形式，可以是数值形式"Evaluate"，也可以是字符串形式"literal"。对于参数"a"，"Prompt"中填写"Parameters A"，在编辑框"Variable"中，填入"a"，在"Control"中，选择"Edit"，在"Assignment"中，选择"Evaluate"。其他的参数(b，D)设置过程与此类似。

第二个延迟模块的封装过程基本类似，只是参数不同。它只有两个参数：增益 a 和延迟量 D。创建相应的子系统并加以保存。

有了这两个子系统。就可以通过对他们的适当组合来生成数字回响滤波器。具体的组合方式多种多样，并随着相对与声源不同的位置而不同。下面是一个对实际效果模拟比较逼真的组合方案。它的框图如图 3-12 所示。

该系统中，左面的四个子系统是第一种延迟相加模块，他们的参数分别为

① a=0.75，b=1，D=29

② a=0.75，b=0.9，D=37

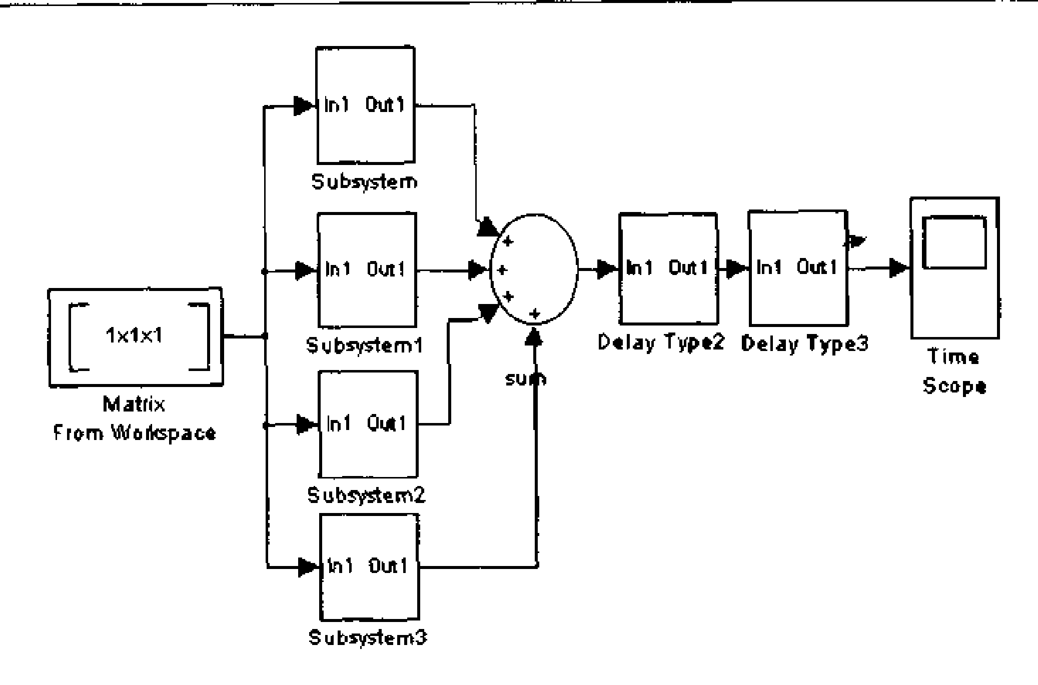


图 3-12 数字音频回响处理器框图

③ a=0.75，b=0.8，D=44

④ a=0.75，b=0.7，D=50

右面的两个子系统是第二种延迟相加模块，他们的参数分别为

① a=0.75，D=27

② a=0.75，D=31

输入信号是一个从主命令空间定义的矩阵，双击输入信号对应的模块，设置相应的属性：由于它对应的输入信号是脉冲函数，因此它的“3D arry”属性应设为 1，即只有一个脉冲；它的 “Sample Time” 属性可以设置为 0.5。

输出信号直接送到“Scope”示波器，结果如图 3-13 所示。

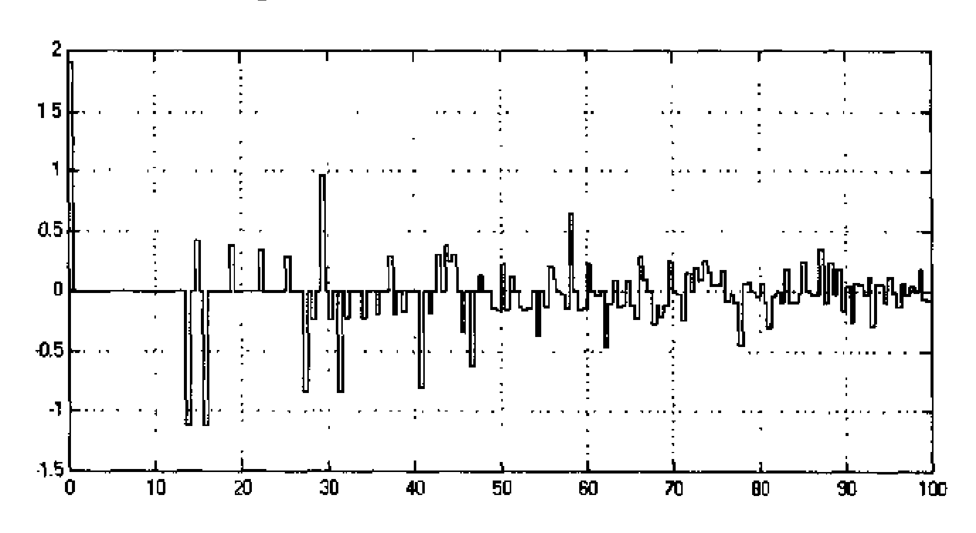


图 3-13 数字音频回响处理器的单位脉冲响应

从图 3-13 可以看出，所设计的数字音频回响处理系统确实能反映出音乐厅的回响性能：输入一个单位脉冲，输出信号在零时刻取值代表直接到达的声音，过一表间隔后的取值代表前期的反射声音信号，再往后杂乱密集的取值代表后期的反射声音信号。

3.1.6 双音多频信号的检测与分析

DTMF 是按键电话通信的德国名称，在 DTMF 通信系统中，高频音与低频音的一个组合表示一个特定的数字或字符*和#。8 个频率按图 3-14 分配共提供 16 个字符，其中 12 个分配如图 3-14 所示，而其他 4 个保留以备进一步扩展用。

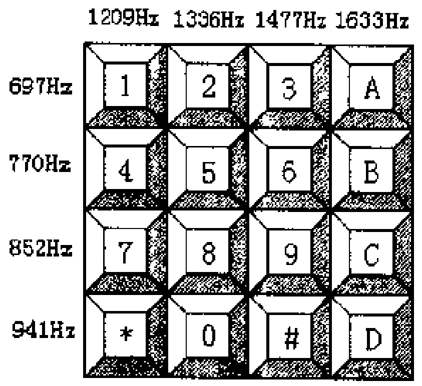


图 3-14 DTMF 数字

DTMF 信号容易用软件产生并用调谐在这 8 个音频的数字滤波器检测，也可用软件实现检测。通常借助编译码器芯片或线性 A/D 和 D/A 变换器将 DTMF 信号接到模拟域，编译码器包含作为双向模/数接口所必须的 A/D 和 D/A ，采样和滤波电路。

DTMF 音频可以用数学方法或查表法产生。在硬件实现中，两个正弦波的数字样本用数学方式产生，并按比例相加在一起，将和信号进行对数压缩并送到编译码器变换成模拟信号。在 8kHz 的采样频率上，硬件必须每 125ms 输出一个样本，在这种情况下，正弦波形的值可以迅速地计算出来，可以节省查表法所要求的大容量存贮空间。

在接收端，收到的对数压缩的 8-位数据码，经对数扩张为 16-位线性码形式，然后检测音频频率来确定所发送的数字。检测算法可以是用 FFT 算法的 DFT，或是用一组滤波器实现。当要检测的音频数目比较少时，滤波器组实现更有效。下面用 Goertzel 算法实现 8 个调谐滤波器。

N 点数据 $x(n)$ 的 DFT 为

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk} \quad k=0,1,\cdots,N-1 \tag{3-1}$$

利用 W_N^k 的周期性和 $W_N^{-kN}=1$，可将 DFT 运算表示为线性滤波运算，即

$$X(k)=W_N^{-kN}X(k)=\sum_{m=0}^{N-1}x(m)W_N^{-k(N-m)} \tag{3-2}$$

如果定义序列 $y_k(n)$ 为

$$y_k(n)=\sum_{m=0}^{N-1}x(m)W_N^{-k(n-m)} \tag{3-3}$$

则 $y_k(n)$ 就是长度为 N 的有限长输入序列 $x(n)$ 与具有如下单位脉冲响应

$$h_k(n)=W_N^{-kn}u(n) \tag{3-4}$$

的滤波器的卷积，该滤波器在 $n=N$ 点的输出就是 DFT 在频点 $\omega_k=\frac{2\pi}{N}k$ 的值，即

$$X(k)=y_k(n)|_{n=N} \tag{3-5}$$

单位脉冲响应为 $h_k(n)$ 的滤波器的系统函数为

$$H_k(z)=\frac{1}{1-W_N^{-k}z^{-1}} \tag{3-6}$$

这个滤波器只有一个位于单位圆上频率为 $\omega_k=\frac{2\pi}{N}k$ 处的极点。因此，可以使输入数据块通过 N 个并行的单极点滤波器（谐振器）组来计算全部 DFT，其中每个滤波器有一个位于 DFT 相应频率的极点。

用式（3-6）给出的滤波器差分方程以迭代方法计算 $y_k(n)$，来代替式（3-3）卷积计算 DFT，于是有

$$y_k(n) = W_N^{-k} y_k(n-1) + x(n) \quad , y_k(-1) = 0 \tag{3-7}$$

预期的输出为 $X(k) = y_k(N)$，为了执行该计算，可以只计算一次相位因子 W_N^{-k}，将其存贮起来。

将具有一对共轭极点的谐振器组合在一起，得到实数型具有如下形式系统函数的双极点滤波器：

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} \tag{3-8}$$

该系统可用图 3-15 表示，实现的差分方程为

$$v_k(n) = 2\cos\frac{2\pi k}{N} v_k(n-1) - v_k(n-2) + x(n) \tag{3-9}$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \tag{3-10}$$

初始条件为 $v_k(-1) = v_k(-2) = 0$。这就是 Goertzel 算法。

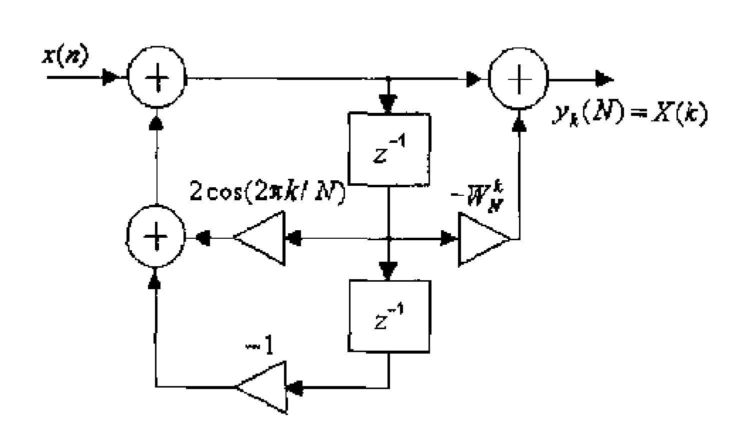


图 3-15 计算 DFT 的双极点谐振器实现

式（3-9）中的递推关系对 n=0,1,…,N 重复 N+1 次，但方程（3-10）仅在 $n = N$ 时刻计算一次，每次计算只需要计算一次实数乘和两次实数加。所以对实数序列 $x(n)$ 由于对称性，用这种算法求出 $X(k)$ 和 $X(N-k)$ 的值需要 N+1 次实数乘法运算。

我们现在能够用 Goertzel 算法实现 DTMF 解码器了。由于有 8 种可能的音频要检测，所以需要 8 个式（3-8）所给出的滤波器，每个滤波器调谐到这 8 个频率之一上。在 DTMF 解码器中，不需要计算复数值 $X(k)$，只要求出幅度值 $|X(k)|$ 或幅度平方值 $|X(k)|^2$ 就够了。所以，计算涉及分子项(滤波器计算的前向部分)的 DFT 值的最后一步可以简化，则有

$$\begin{aligned} |X(k)|^2 &= |y_k(N)|^2 = |v_k(N) - W_N^k v_k(N-1)|^2 \\ &= v_k^2(N) + v_k^2(N-1) - \left(2\cos\frac{2\pi k}{N}\right) v_k(N) v_k(N-1) \end{aligned} \tag{3-11}$$

于是 DTMF 解码器中就完全避免了复数运算。

[例 6] 设计 DTMF 拨号：本题的目的是理解 DTMF 音频产生软件和 DTMF 解码算法(Goertzel 算法)，设计如下 MATLAB 程序模块。

① 音频产生函数　它接收电话拨号数列，产生包含适当音频(见图 3-14)的信号，每个数据信号持续半秒，采样频率为 8kHz。

② 拨号音频产生器　它在规定的持续时间内以 8kHz 的采样频率，产生（350+440）Hz 频率的样本。

③ 实现式（3-11）的解码函数　它接收 DTMF 信号，产生包含拨号数字的数列。

用 MATLAB 产生包含数字和拨号音频的几个拨号表矩阵，做音频产生和检测程序实验，用 MATLAB 的声音生成功能监听产生的音频，并观测所产生音频的成分。如图 3-16 所示。MATLAB程序实现如下：

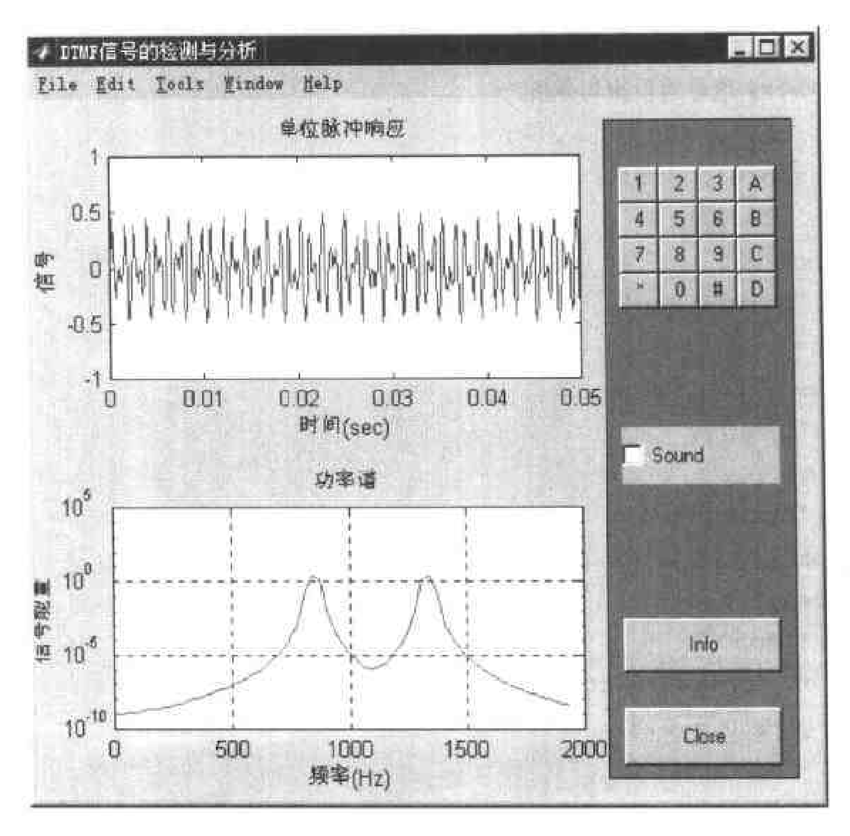


图 3-16　DTMF 信号的分析与检测界面

```
function phone1(action)
%phone1  Signal processing and the touch-tone phone1.
persistent Fs tones t;
if isempty(tones),
   % Generate DTMF tones:
   Fs  = 8000;
   t   = (0:799)/Fs;    % 800 samples at Fs
   pit = 2*pi*t;
   fc = [697 770 852 941];
   fr = [1209 1336 1477 1633];
   f  = [];
   for c=1:4,
      for r=1:4,
         f = [ f [fc(c);fr(r)] ];
      end
   end
   for i=1:size(f,2),
      tones(:,i) = sum(sin(f(:,i)*pit))'/4;
   end
end
```

```
if nargin<1,
   action='initialize';
end
if strcmp(action,'initialize'),
   figNumber=figure( ...
      'Name','DTMF 信号的检测与分析', ...
      'NumberTitle','off', ...
      'Backingstore','off', ...
      'Visible','off');
   axHndl1=axes( ...
      'Units','normalized', ...
      'Position',[0.10 0.60 0.60 0.32], ...
      'Drawmode','fast', ...
      'Visible','on');
   axHndl2=axes( ...
      'Units','normalized', ...
      'Position',[0.10 0.10 0.60 0.32], ...
      'Drawmode','fast', ...
      'Visible','on');
   % Information for all buttons
   labelColor=[0.8 0.8 0.8];
   top=0.85;
   left=0.75;
   bottom=0.05;
   xLabelPos=0.75;
   labelWid=0.20;
   labelHt=0.05;
   btnWid=0.20;
   btnHt=0.08;
   % Spacing between the button and the next command's label
   spacing=0.05;
   btnOffset=0;
   % The CONSOLE frame
   % Use the frame's userdata to hold tbe tones matrix
   frmBorder=0.02;
   yPos=bottom-frmBorder;
   frmPos=[left-frmBorder yPos btnWid+2*frmBorder 0.9+2*frmBorder];
   frmHndl=uicontrol( ...
      'Style','frame', ...
      'Units','normalized', ...
      'Position',frmPos, ...
      'BackgroundColor',[0.50 0.50 0.50]);
   % The TONE buttons
   %btnNumber=1;
```

```
%yLabelPos=0.90-(btnNumber-1)*(btnHt+labelHt+spacing);
labelStr=' Button No.';
    for count=1:16,
    btnPos=[xLabelPos+rem(count-1,4)*(btnWid/4) ...
            top-floor((count-1)/4)*(btnWid/4) btnWid/4 btnWid/4];
        s = '123A456B789C*0#D';
        btnStr = s(count);
    uicontrol( ...
        'Style','pushbutton', ...
        'Units','normalized', ...
        'Position',btnPos, ...
        'String',btnStr, ...
        'UserData',count, ...
        'Callback','phone1(''tone'')');
end
% The SOUND button
btnNumber=2;
yPos=top-4*btnHt;
labelStr='Sound';
% Setting this checkbox will allow the sound to work
% Generic button information
btnPos=[left yPos-btnHt btnWid btnHt];
sndHndl=uicontrol( ...
    'Style','checkbox', ...
    'Units','normalized', ...
    'Position',btnPos, ...
    'Enable','on', ...
    'String',labelStr);
% The INFO button
labelStr='Info';
callbackStr='phone1(''info'')';
infoHndl=uicontrol( ...
    'Style','push', ...
    'Units','normalized', ...
    'Position',[left bottom+btnHt+spacing btnWid btnHt], ...
    'String',labelStr, ...
    'Callback',callbackStr);
% The CLOSE button
labelStr='Close';
callbackStr='close(gcf)';
closeHndl=uicontrol( ...
    'Style','push', ...
    'Units','normalized', ...
    'Position',[left bottom btnWid btnHt], ...
    'String',labelStr, ...
```

```
        'Callback',callbackStr);
    % Initialize the plot with tone number 1
    axes(axHndl1);
    tone=tones(:,1);
    plot1Hndl=plot(t,tone,'Erasemode','background');
    xlabel('时间(sec)');
    ylabel('信号');
    title('单位脉冲响应');
    set(axHndl1,'XLim',[0 0.05],'YLim',[-1 1]);
    axes(axHndl2);
    p = psd(tones(:,1),256,[],[],128);
    f = (0:127)/128 * (Fs/2);
    f=f(1:63);
    p=p(1:63,1);
    plot2Hndl=semilogy(f,p,'Erasemode','background');grid;
    xlabel('频率(Hz)');
    ylabel('信号能量');
    title('功率谱');
    set(axHndl2,'XLim',[0 2000]);
    % Uncover the figure
    set(figNumber,'Visible','cn', ...
        'UserData',[frmHndl infoHndl closeHndl plot1Hndl plot2Hndl sndHndl]);
elseif strcmp(action,'tone'),
    figNumber=watchon;
    hndlList=get(figNumber,'Userdata');
    frmHndl=hndlList(1);
    infoHndl=hndlList(2);
    closeHndl=hndlList(3);
    plot1Hndl=hndlList(4);
    plot2Hndl=hndlList(5);
    sndHndl=hndlList(6);
    %set([closeHndl infoHndl],'Enable','off');
    % Playback the tone here
    % Sampling rate is Fs Hz
    toneChoice=get(gco,'UserData');
    tone=tones(:,toneChoioe);
    if get(sndHndl,'Value'),
        sound(tone,Fs);
    end
    p = psd(tone,256,[],[],128);
    f = (0:127)/128 * (Fs/2);
    p=p(1:63,1);
    set(plot1Hndl,'YData',tone);
    set(plot2Hndl,'Ydata',p);
    %End of playing tone
```

```
    %set([closeHndl infoHndl],'Enable','on');
    watchoff(figNumber);
elseif strcmp(action,'info'),
    ttlStr=get(gcf,'Name');
    hlpStr= ...
        [' This window demonstrates the speed and     '
         ' utility of the "psd" command in the        '
         ' Signal Processing Toolbox.                 '
         ' The touch tone phone pad in the upper right'
         ' plays the actual tones used by a normal    '
         ' phone. The upper plot shows a sample of the'
         ' time response of the sound. The lower plot '
         ' shows the spectrum associated with the waveform. '
         ' Notice that two tones are used to code each'
         ' number on the dial.                        '
         ' File name:phone1.m                         '];
    helpfun(ttlStr,hlpStr);
end
% if strcmp(action, ...
% [EOF] phone1.m
```

用 simulink 实现的 DTMF 信号发生器如图 3-17 所示。

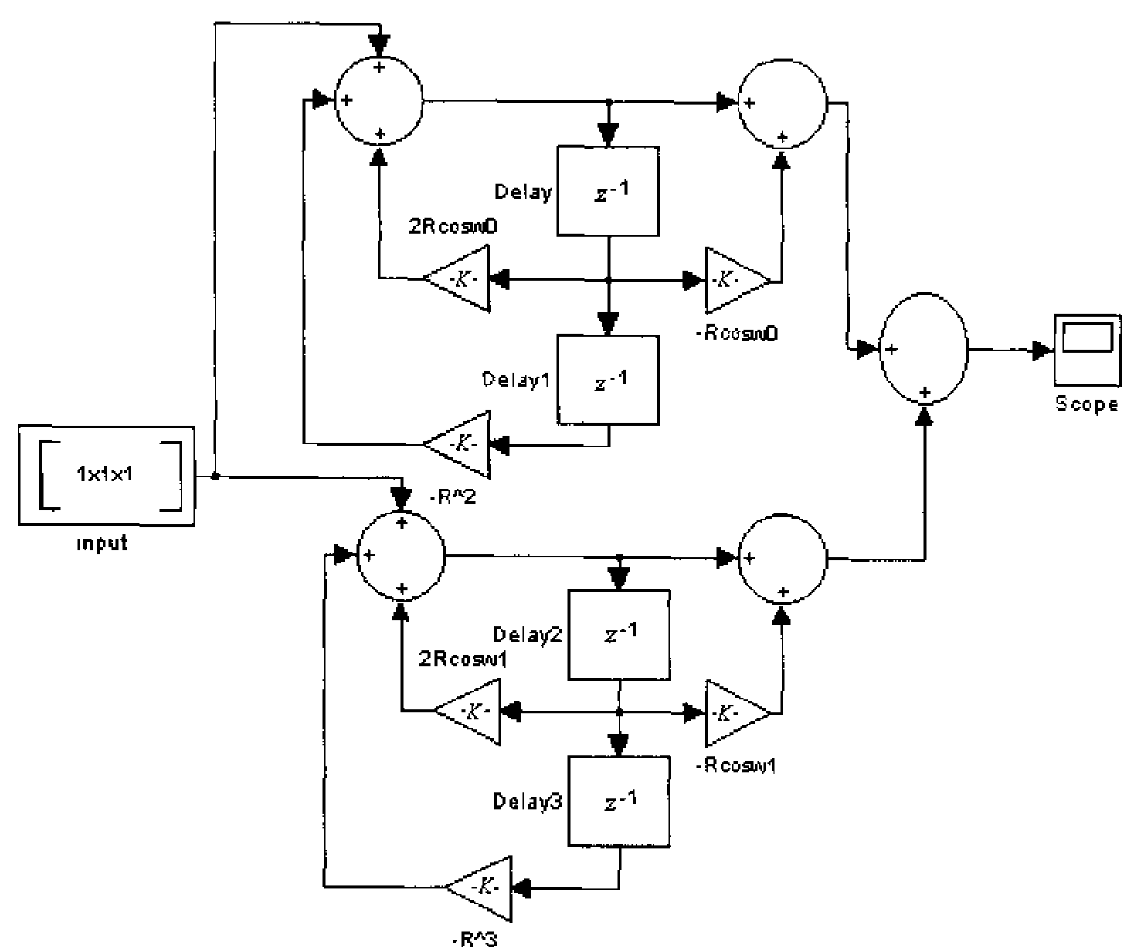


图 3-17　DTMF 音频发生器

在 MATLAB 命令窗口输入：R=1;w0=852*2*pi/8000;w1=1209*2*pi/8000;然后进行仿真，在示波器上可以观察到 DTMF 信号“7”的单位脉冲响应如图 3-18 所示。

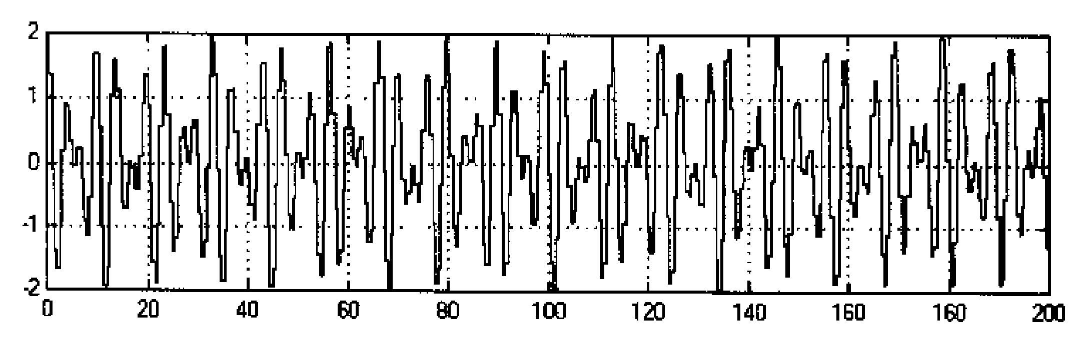


图 3-18　DTMF 信号“7”的单位脉冲响应

3.1.7　调制电路的软件实现

在数字信号通信快速发展以前主要是模拟通信，由于为了合理使用频带资源，提高通信质量，需要使用模拟调制技术，通常连续波的模拟调制是以正弦波为载波的调制方式，它分为线性调制和非线性调制。线性调制是指调制后的信号频谱为调制信号频谱的平移或线性变换，而非线性调制则没有这个性质。

3.1.7.1　常规双边带调幅（AM）的实现

在常规双边带调幅中，载波的幅度包络与输入的调制信号成正比，其时域表达式为

$$S_{\mathrm{AM}}(t)=[A_0+m_{\mathrm{a}}f(t)]\cos(\omega_{\mathrm{c}}t+\theta_{\mathrm{c}})$$

式中　A_0——外加的直流分量；

$f(t)$——调制信号，可以是确定信号，也可以是随机信号；

m_{a}——调制系数，它的范围在（0，1）之间；

ω_{c}——载波的角频率；

θ_{c}——载波的初始相位。

若记 $F(f)$ 为调制信号的频域表达式，则已调信号的频域表达式为

$$S_{\mathrm{AM}}(f)=\frac{A_0}{2}\delta(f-f_{\mathrm{c}})+\frac{A_0}{2}\delta(f+f_{\mathrm{c}})+\frac{1}{2}F(f-f_{\mathrm{c}})+\frac{1}{2}F(f+f_{\mathrm{c}})$$

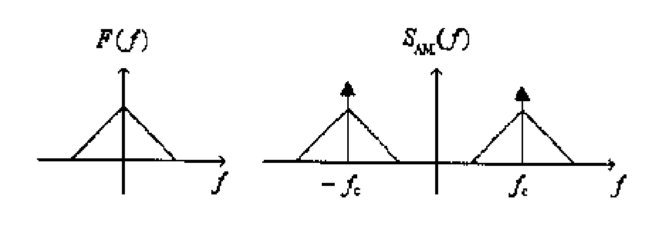


图 3-19　常规幅度调制的频谱图

从频域表达式可以看出，已调信号的频带宽度是调制信号的频带的两倍，此种调制方式占用频带较宽。由于被调制信号的包络就是调制信号叠加一个直流，因此易于实现峰值包络解调。从频域表达式中可以看出包含有正弦载波分量，即有部分功率消耗在载波上，没有用于信息的传递，从效率上看，常规双边带调幅方式效率较低，但由于其调制和解调都简单，因而得到广泛的应用。典型的常规双边带调幅调制的频谱图如图 3-19 所示。

[例 7]　一有限长度信号，其时域表达式为

$$S(t)=\begin{cases} t & 0<t<t_0/4 \\ -t+t_0/4 & t_0/4<t<3t_0/4 \\ t-t_0 & 3t_0/4<t<t_0 \end{cases}$$

将其调制在载波 $C(t)=\cos(2\pi f_c t)$ 上，假设 $t_0=0.5\text{s}, f_c=50\text{Hz}$，调制系数为 $m_a=0.8$。求出已调制信号的时域表达式及时域波形，未调信号和已调信号的频谱关系图，计算未调信号和已调信号的功率，在有噪声的情况下假设信噪比为10dB的噪声功率。

（1）已调制信号的时域表达式

$$M(t)=[1+0.8S(t)/0.125]\cos 2\pi f_c t$$

给 $S(t)$ 乘 $1/0.125$ 是因为调制时要进行归一化。图 3-20 表示的是三角脉冲信号、载波信号和已调制的信号，可以看出已调信号的包络就是未调信号，所以可以用已调制信号的峰值进行解调。

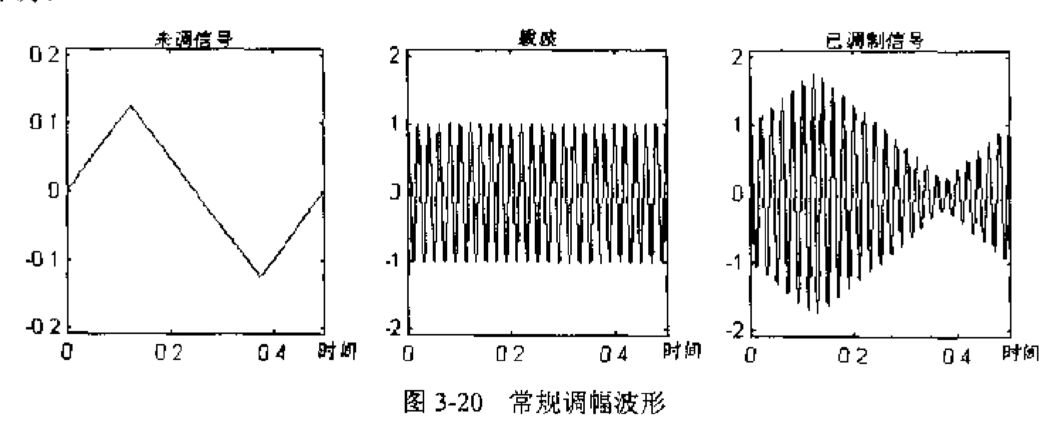


图 3-20 常规调幅波形

（2）未调信号和已调信号的频谱关系图

未调信号的频谱频率分量主要集中在低频但直流并不多，经调制后，频谱被搬移到载波的频率附近，并出现了较多的载频分量。原因是调制时采用了给未调信号叠加直流，而直流调频后成为载频，如图 3-21 所示。

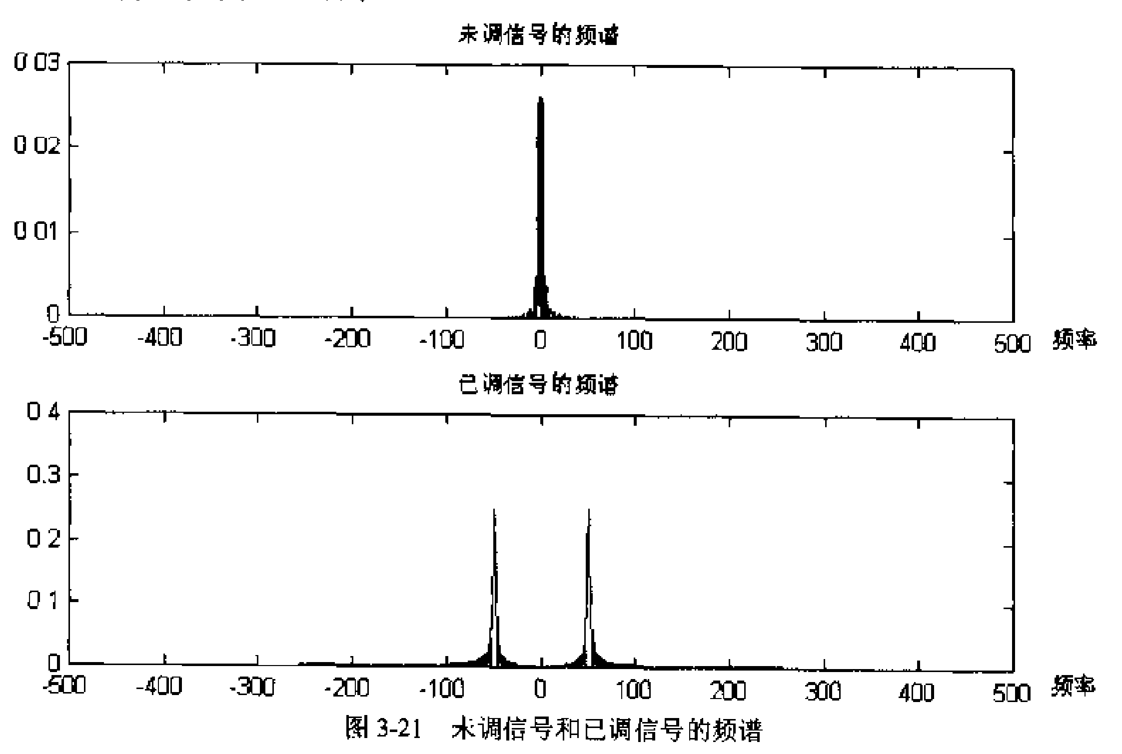


图 3-21 未调信号和已调信号的频谱

（3）未调信号和已调信号的功率计算

未调信号的功率

$$P_S = \frac{1}{0.5}\left[\int_0^{0.125} t^2 \mathrm{d}t + \int_{0.125}^{0.375}(0.25-t)^2 \mathrm{d}t + \int_{0.375}^{0.5}(t-0.5)^2 \mathrm{d}t\right] = 0.0052$$

归一化功率为　　$P_{S_n} = P_S/(0.125)^2 = 0.3327$

调制效率为　　$\eta = \dfrac{a^2 P_{S_n}}{1+a^2 P_{S_n}} = 0.1755$

可以看出常规调幅调制的效率是比较低的。

调制信号的功率为　　$P_m = E[1+a\,m_n(t)]/2 = 0.6074$

（4）在给定的信噪比10dB条件下，噪声功率为

$$P_n = \eta P_m / 10 = 0.0107$$

图 3-22 为噪声和叠加了噪声的已调制信号的时域波形。

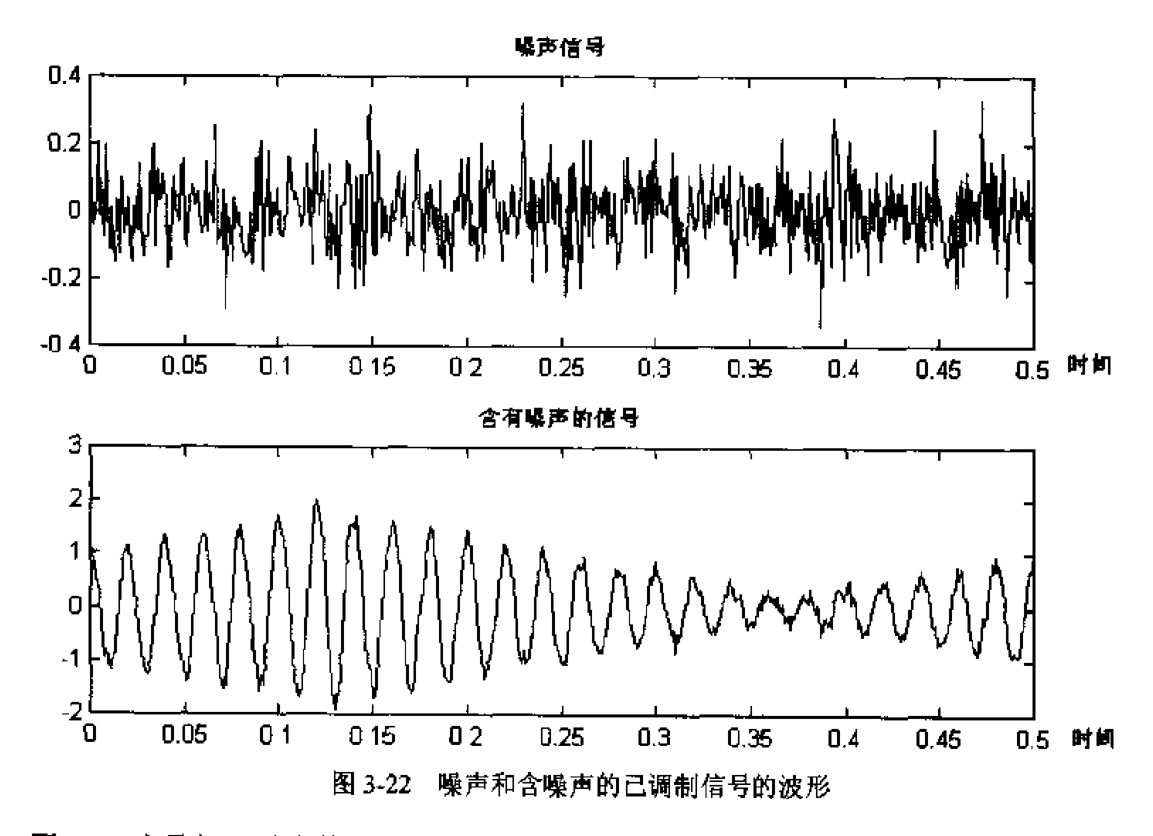


图 3-22　噪声和含噪声的已调制信号的波形

图 3-23 为叠加了噪声的已调制信号的频谱。

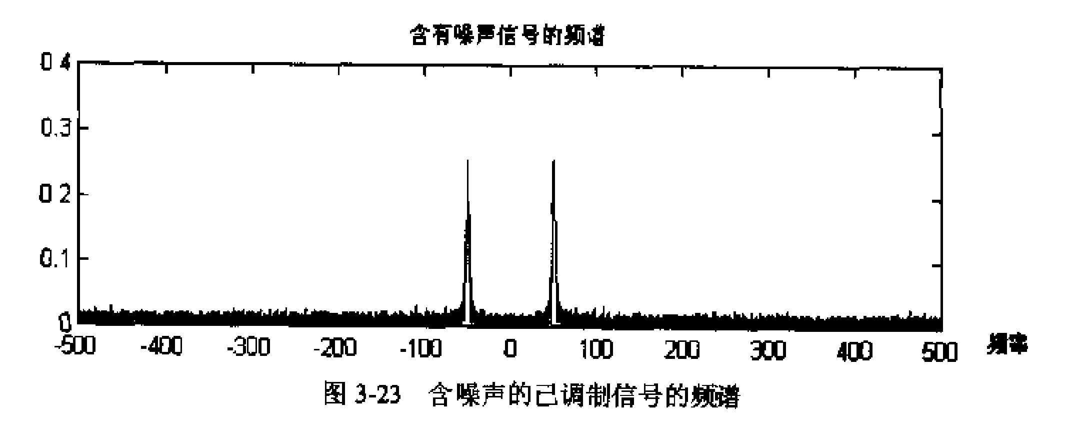


图 3-23　含噪声的已调制信号的频谱

上面的计算是用下面程序实现的。

```
%amodulate.m
df=0.2; %频率分辨率
t0=0.5; %定义 t0 信号的持续时间的值
tz=0.001; %定义采样时间
fc=50; %定义载波频率
snr=10; %定义信噪比，用 dB 表示
a=0.8; %定义调制系数
snr_lin=10^(snr/10); %信噪比的数值
fz=1/tz; %定义采样频率
t=0:tz:t0; %定义采样点数据
m=zeros(1,501); %定义信号
for i=1:1:125
  m(i)=i;
end
for i=126:1:375
  m(i)=m(125)-i+125;
end
for i=376:1:501
  m(i)=m(375)+i-375;
end
m=m/1000;
c=cos(2*pi*fc.*t);%载波信号
m_n=m/max(abs(m));
[M,m,df1]=fftseq(m,tz,df);%傅里叶变换
M=M/fz;%频率缩放，便于作图
u=(1+a*m_n).*c;%将调制信号调制在载波上
[U,u,df1]=fftseq(u,tz,df);%对已调制信号作傅里叶变换
U=U/fz;%频率缩放，便于作图
f=[0:df1:df1*(length(m)-1)]-fz/2; %定义频率向量
signal_power=ampower(u(1:length(t))) %计算已调制信号的功率
pmn=ampower(m(1:length(t)))/(max(abs(m)))^2 %计算未调制信号的功率
eta=(a^2*pmn)/(1+a^2*pmn) %计算调制效率
noise_power=eta*signal_power/snr_lin %计算噪声功率
noise_std=sqrt(noise_power) %噪声标准差
noise=noise_std*randn(1,length(u)); %产生高斯分布的噪声
r=u+noise; %总接收信号
[R,r,df1]=fftseq(r,tz,df); %总接收信号傅里叶变换
R=R/fz; %频率缩放
noise=noise_std*randn(1,length(u));
r=u+noise;
[R,r,df1]=fftseq(r,tz,df);
R=R/fz;pause
subplot(1,3,1);plot(t,m(1:length(t)));
```

```
axis([0,0.5,-0.21,0.21]);xlabel('时间');title('未调信号');
subplot(1,3,2);plot(t,c(1:length(t)));
axis([0,0.5,-2.1,2.1]);xlabel('时间');title('载波');
subplot(1,3,3);plot(t,u(1:length(t)));
axis([0,0.5,-2.1,2.1]);xlabel('时间');title('已调制信号');pause
subplot(2,1,1);plot(f,abs(fftshift(M)));
xlabel('频率');  title('未调信号的频谱');
subplot(2,1,2);plot(f,abs(fftshift(U)));
title('已调信号的频谱');xlabel('频率');pause
subplot(3,1,1);plot(t,noise(1:length(t)));
title('噪声信号');xlabel('时间');
subplot(3,1,2);plot(t,r(1:length(t)));
title('含有噪声的信号'); xlabel('时间');
subplot(3,1,3);plot(f,abs(fftshift(R)));
title('含有噪声信号的频谱');xlabel('频率');
%===================================
function[M,m,df]=fftseq(m,tz,df)
 fz=1/tz;
 if nargin==2 %判断输入参数的个数是否符合要求
 n1=0;
 else n1=fz/df; %根据参数个数决定是否使用频率缩放
 end
 n2=length(m);
 n=2^(max(nextpow2(n1),nextpow2(n2)));
 M=fft(m,n); %进行离散傅里叶变换
 m=[m,zeros(1,n-n2)];
 df=fz/n;
%===================================
function p=ampower(x) %此函数仅用于计算本例的信号功率
  p=(norm(x)^2)/length(x); %计算信号的能量
  t0=0.5;
  tz=0.001;
  m=zeros(1,501);
  for i=1:1:125%计算第一段信号的功率
     m(i)=i;
  end
  for i=126:1:375 %计算第二段信号的功率
     m(i)=m(125)-i+125;
  end
  for i=376:1:501 %计算第三段信号的功率
     m(i)=m(375)+i-375;
  end
  m=m/1000; %功率归一化
  m_hat=imag(hilbert(m));
```

3.1.7.2 抑制载波双边带调幅（DSB）的实现

由于常规调幅调制的效率太低，耗用了大量功率，在小功率场合很不方便，而抑制载波双边带调幅就克服了效率低的缺点。它的时域表达式为

$$S_{\rm DSB}(t)=Af(t)\cos(\omega_{\rm c}t+\theta_{\rm c})$$

若 $F(f)$ 为调制信号的频域表达式，已调信号的频域表达式为

$$S_{\rm DSB}(f)=\frac{1}{2}F(f-f_{\rm c})+\frac{1}{2}F(f+f_{\rm c})$$

从频域表达式可以看出，已调信号的频带宽度仍是调制信号的频带的两倍。典型的抑制载波双边带调幅调制的频谱图如图 3-24 所示。

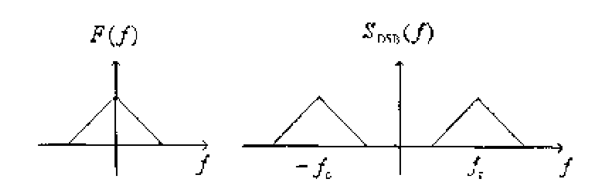


图 3-24 抑制载波幅度调制的频谱图

[例 8] 一个未调信号 $f(t)=\begin{cases}\text{sinc}(200t) & |t\leqslant t_0\\ 0 & 其他\end{cases}$

$t_0=2\text{s}$，载波为 $\cos 2\pi f_{\rm c}t$，$f_{\rm c}=200\text{Hz}$，用抑制载波调幅来调制信号，给出调制信号 $M(t)$ 波形，画出未调信号和调制信号的频谱。

① 调制信号 $M(t)=\begin{cases}\text{sinc}(200t)\cos(400\pi t) & |t|\leqslant 2\\ 0 & 其他\end{cases}$

调制信号的波形如图 3-25 所示。

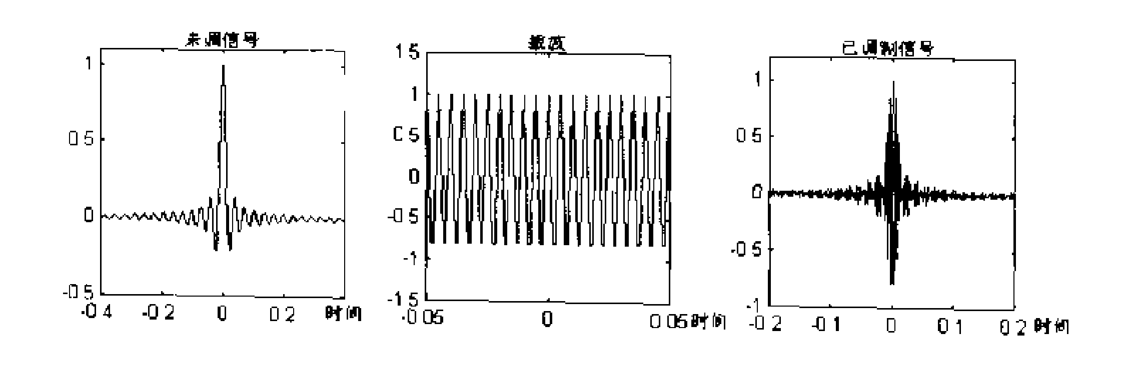


图 3-25 抑制载波调幅波形

② sinc 函数的频谱应该是一个矩形波样的谱，但由于 $f(t)$ 只是 sinc 函数的一段，并且在计算机上使用离散的数字来存储，因为计算精度的要求，使得频谱与矩形波形谱不完全一样。频谱图如图 3-26 所示。

③ MATLAB 实现抑制载波调幅程序如下。

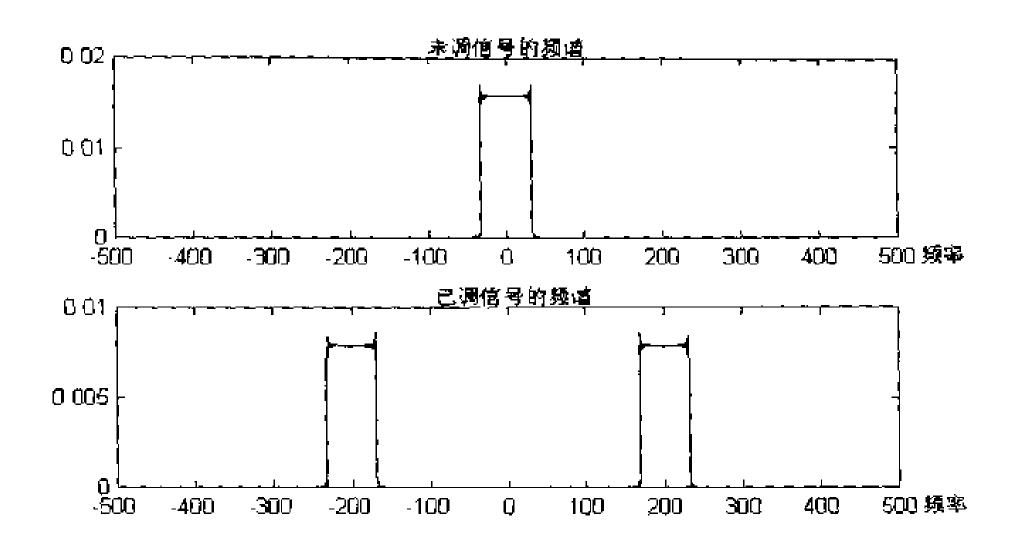


图 3-26 抑制载波调幅频谱

```
%dsbmod.m
df=0.3;
t0=2;
tz=0.001;
fc=200;
fz=1/tz;
t=[-t0/2:tz:t0/2];
x=sin(200*t);
m=x./(200*t+eps);
m(1001)=1;
c=cos(2*pi*fc.*t);
u=m.*c;
[M,m,df1]=fftseq(m,tz,df);
M=M/fz;
[U,u,df1]=fftseq(u,tz,df);
U=U/fz;
f=[0:df1:df1*(length(m)-1)]-fz/2;pause
subplot(1,3,1);plot(t,m(1:length(t)));
axis([-0.4,0.4,-0.51,1.1]);xlabel('时间');title('未调信号');
subplot(1,3,2);plot(t,c(1:length(t)));
axis([-0.05,0.05,-1.5,1.5]);xlabel('时间');title('载波');
subplot(1,3,3);plot(t,u(1:length(t)));
axis([-0.2,0.2,-1,1.2]);xlabel('时间');title('已调制信号');pause
subplot(2,1,1);plot(f,abs(fftshift(M)));
xlabel('频率');title('未调信号的频谱');
subplot(2,1,2);plot(f,abs(fftshift(U)));
title('已调信号的频谱');xlabel('频率');
```

3.1.7.3 单边带调幅（SSB）的实现

常规双边带调幅和抑制载波双边带调幅都有一个缺点就是，占用频带资源过宽，从频谱

图上看有一半的频带资源被浪费了。为此提出单边带调幅调制的方式，此种方式的优点是，调制效率高，占用频带资源相对较小。

单边带调幅方式的时域表达式比较复杂，有上边带（USB）和下边带（LSB）两种方式，时域表达式分别为

$$S_{\mathrm{USB}}(t)=\frac{1}{2}f(t)\cos\omega_c t-\frac{1}{2}f(\hat{t})\sin\omega_c t$$

$$S_{\mathrm{LSB}}(t)=\frac{1}{2}f(t)\cos\omega_c t+\frac{1}{2}f(\hat{t})\sin\omega_c t$$

式中 $f(\hat{t})$ 为 $f(t)$ 的 Hilbert 变换，即

$$f(\hat{t})=f(t)*\frac{t}{\pi}=\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{f(u)}{t-u}\mathrm{d}u$$

$$f(t)=-f(\hat{t})*\frac{t}{\pi}=-\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{f(\hat{u})}{t-u}\mathrm{d}u$$

式中*表示卷积，频域上两者的关系为

$$F(\hat{\omega})=-\mathrm{j}\,\mathrm{sgn}(\omega)F(\omega)$$

$$F(\omega)=\mathrm{j}\,\mathrm{sgn}(\omega)F(\hat{\omega})$$

若 $F(f)$ 为调制信号的频域表达式，单边带已调信号的频域表达式为

$$S_{\mathrm{USB}}(f)=F(f-f_c)+F(f+f_c) \qquad f_c\leqslant|f|$$

$$S_{\mathrm{LSB}}(f)=F(f-f_c)-F(f+f_c) \qquad f_c\geqslant|f|$$

上边带的频谱图如图 3-27 所示。

下边带的频谱图如图 3-28 所示。

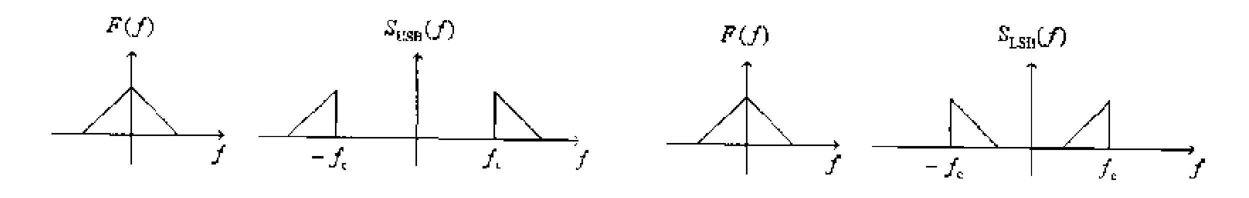


图 3-27　上边带调幅波的频谱图　　图 3-28　下边带调幅波的频谱图

从图 3-27 和图 3-28 可以看出，已调信号的频带宽度就是调制信号的频带宽度。

[例 9]　一个未调信号 $f(t)=\begin{cases}\mathrm{sinc}(200t) & |t|\leqslant t_0\\ 0 & \text{其他}\end{cases}$

t_0=2s，载波为 $\cos 2\pi f_c t$，f_c=100Hz，用单边带调幅来调制信号，给出调制信号 $M(t)$波形，画出未调信号和调制信号的频谱。

调制信号波形如图 3-29 所示，从时域波形可以看出单边带调幅波形的包络已经不能反映未调信号的幅度了，所以只能采用相干解调方式。上下边带的调制波形区别不是很大，因为他们的频谱具有某种对称性。

变换前后的频谱图如图 3-30 所示。

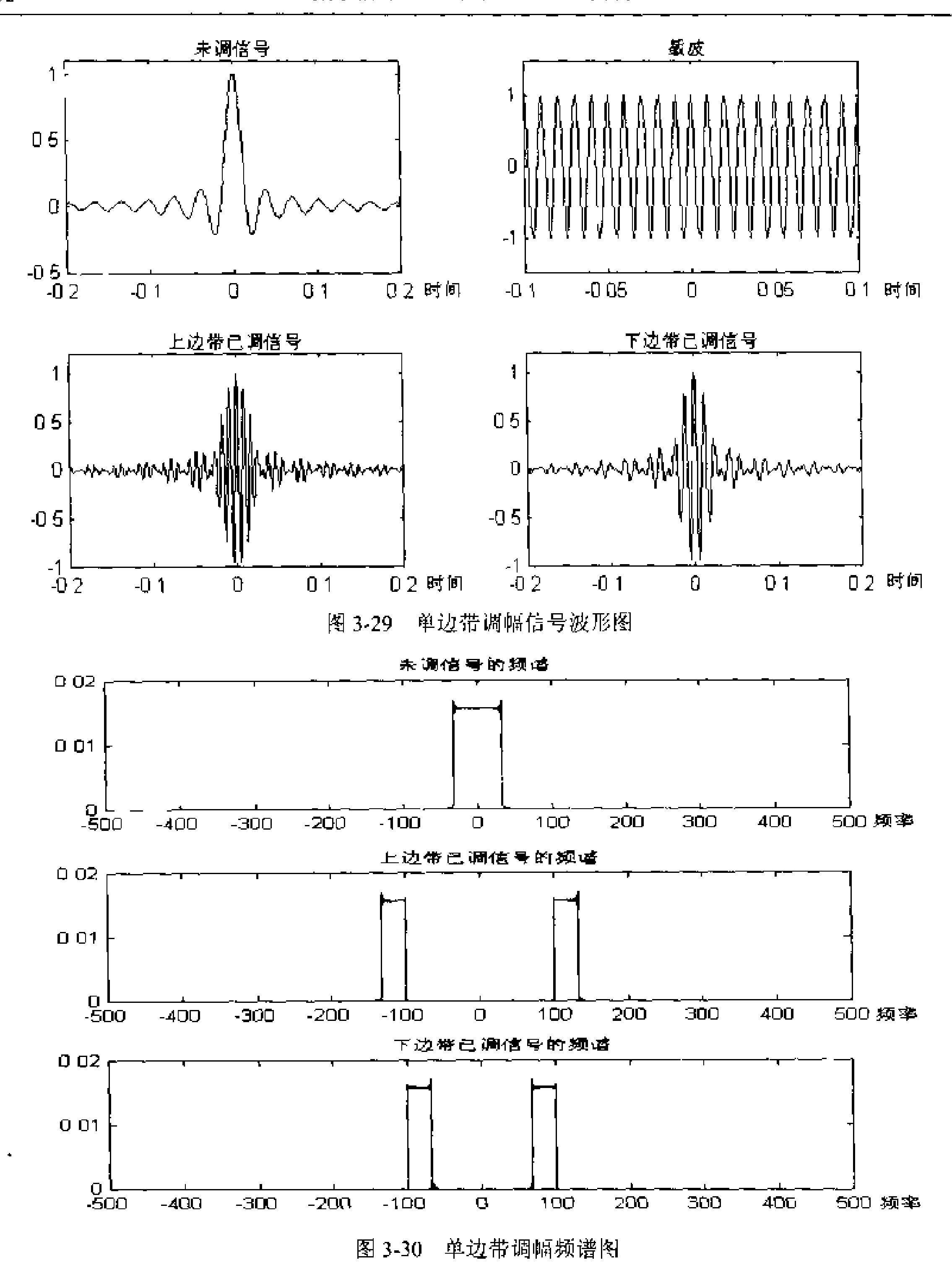


图 3-29 单边带调幅信号波形图

图 3-30 单边带调幅频谱图

MATLAB 实现单边带调幅程序如下：

```
%sshmod.m
df=0.3; %频率分辨率
t0=2; %信号持续时间
tz=0.001; %采样时间
fc=100; %载波频率
fz=1/tz;
t=[-t0/2:tz:t0/2];
x=sin(200*t); %未调制信号
```

```
m=x./(200*t+eps);
m(1001)=1;
c=cos(2*pi*fc.*t); %载波同向分量
b=sin(2*pi*fc.*t); %载波正交分量
v=m.*c+imag(hilbert(m)).*b; %下边带调幅分量
u=m.*c-imag(hilbert(m)).*b; %上边带调幅分量
[M,m,df1]=fftseq(m,tz,df); %傅里叶变换
M=M/fz;
[U,u,df1]=fftseq(u,tz,df); %傅里叶变换
U=U/fz;
[V,v,df1]=fftseq(v,tz,df); %傅里叶变换
V=V/fz;
f=[0:df1:df1*(length(m)-1)]-fz/2;pause
subplot(221);plot(t,m(1:length(t)));
axis([-0.2,0.2,-0.51,1.1]);xlabel('时间');title('未调信号');
subplot(222);plot(t,c(1:length(t)));
axis([-0.1,0.1,-1.5,1.5]);xlabel('时间');title('载波');
subplot(223);plot(t,u(1:length(t)));
axis([-0.2,0.2,-1,1.2]);xlabel('时间');title('上边带已调信号');
subplot(224);plot(t,v(1:length(t)));
axis([-0.2,0.2,-1,1.2]);xlabel('时间');title('下边带已调信号');pause
subplot(311);plot(f,abs(fftshift(M)));
xlabel('频率');title('未调信号的频谱');
subplot(312);plot(f,abs(fftshift(U)));
title('上边带已调信号的频谱');xlabel('频率');
subplot(313);plot(f,abs(fftshift(V)));
title('下边带已调信号的频谱');xlabel('频率');
```

3.1.8 数字变频技术的应用

频率细化是信号处理中广泛应用的一种技术，它能够提高频率的分辨率，将选定的频率域上的特性曲线放大，从而使系统的频率特性更清楚地显示出来，如图 3-31 所示。

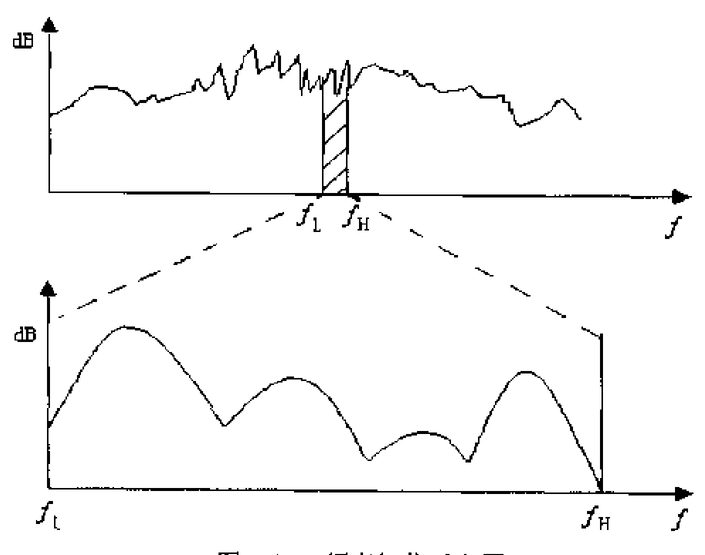


图 3-31 频率细化示意图

设系统的采样频率为f_s，采样点数为N，则频率的分辨率为

$$\Delta f = f_s / N$$

要进行频率细化，即提高频率分辨率，使Δf变小，有两种方法：增加采样点数N和降低采样频率f_s。

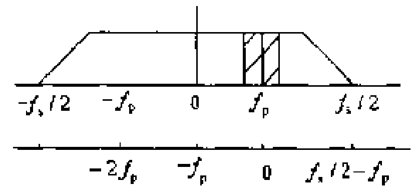


图 3-32 移频后频率刻度

降低采样频率的方法主要依据移频技术，如图 3-32 所示。

设要移频部分的频率为f_p，令

$$h(t) = \mathrm{e}^{-\mathrm{j}\omega_p t}$$

时域信号$x(t)$与$h(t)$乘积后，则在频域上，该信号的f_p频率就移到了原点处。信号频率移到低频后，经过低通滤波，就可以用低的采样频率进行采样，从而达到提高分辨率的目的。具体过程如图 3-33 所示。

移频法频率细化过程中应注意的问题。

① 非细化处理的系统采样频率为f_s，采样点数为N。对于细化过程，设频率细化倍数为M，信号经抗混叠滤波器后进行 A/D 采样，采样频率应仍为f_s，保持不变，采样点数则为$M\times N$，这样就保证了细化与非细化处理的基本频带范围保持不变，并且可以细化这一频带中的任何一段。

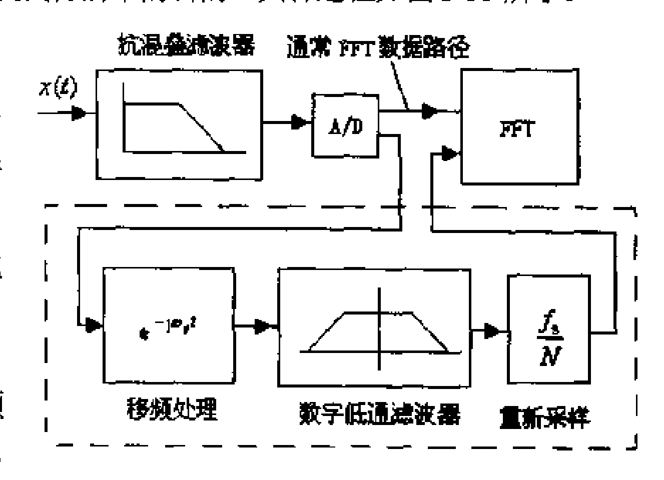


图 3-33 移细法频率细化过程图

② 要细化的频率范围为$f_L\sim f_H$，移频后低频点f_L移到原点，高频点变为(f_H-f_L)。这时数字低通滤波器的截止频率应大于(f_H-f_L)，并小于低通重采样频率f_s/M的一半。即截止频率的范围为$(f_H-f_L)<f_c<f_s/(2M)$，且可以得到最大细化倍数M与细化频率范围之间的关系为$M<f_s/2(f_H-f_L)$。

③ 数字低通滤波器的通带必须平，通带内波动小，这样原信号的频率特性细化后在幅值上才不会改变；滤波器带外衰减大，阻带起始频率小于$f_s/(2M)$，这样就能保证低频重采样时抗混叠的效果最好，细化的效果也最好。

④ 由于采样点数保持不变，因此细化处理的 FFT 时间与不细化的 FFT 时间一样，都是$N\lg(M/2)$；细化过程要进行$M\times N$点的高频采样和N点的低频采样，而不细化过程只进行N点的高频采样，所以在采样时间上细化过程要稍长一点，保与N点 FFT 时间比较可以忽视不计。

⑤ 移频法频率细化与增加采样点数频率细化的比较：移频法频率细化只进行N点的 FFT 变换和一些数组、矩阵运算，所花时间为$N\lg(N/2)$；增加采样点数频率细化要进行$M\times N$点的 FFT 变换，所花时间为：$(M\times N)\lg((M\times N)/2)$。因此移频法频率细化的时间要省得多，这是工程应用中多采用它的原因。

⑥ 由于移频使f_L前面的频段移到频域的负轴上，而低通滤波又滤掉了f_H后面的高频部分，因此这种方法只能进行一段频率的细化，不能进行全频段的细化。要进行全频段的细化，可以采用增加采样点数的方法。

对于信号$x(t)=\sin(2\pi f_1 t)-3\cos(2\pi f_2 t)+4\sin(2\pi f_3 t)$，是由$f_1$=202Hz，$f_2$=204Hz，$f_3$=208Hz

三个不同频率、不同幅值的正余弦信号合成的。采样点数为512，采样频率为5120Hz。则频率的分辨率为10 Hz，在频域内分辨不出这三个信号，如图3-34所示。

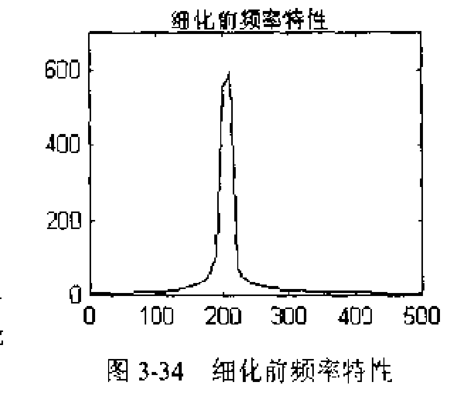


图3-34 细化前频率特性

细化处理需要5120个A/D采样点 $x(n)$，同时将 $h(t)$ 也离散化 $h(n)$，将这两个序列进行乘积，得到一个新的序列，这样就实现了频移过程。

用数字滤波技术设计一个截止频率为20Hz的低通滤波器，将频移后的序列滤波，得到一个只在0～20Hz频段上有特性曲线，其他频段被滤掉的信号序列。

对上面的5120点序列每隔10个点（细化倍数）进行取样，实际是对经频移、滤波后的信号进行重新采样。采样频率为5120Hz/10，即采样频率降低了10倍，抽取得到512个点。

对这512个点进行FFT变换（为使细化前后的频率一致，在作FFT前应反频移），就可以得到200～220Hz频段上已经细化了10倍的频率特性。分别进行8倍、4倍、2倍的频率细化，结果如图3-35所示。细化倍数越大，频率分辨率就越高。

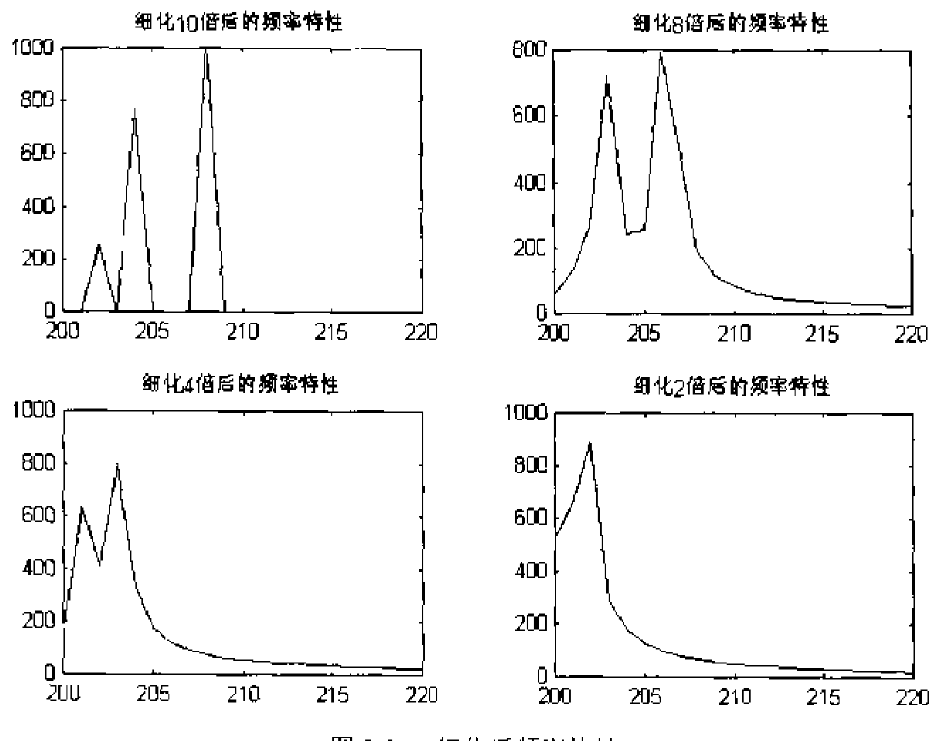


图3-35 细化后频率特性

用MATLAB实现程序如下。

```
%s341.m
t=0:1:5119;Fs=5120;
x=sin(2*pi*202*t/Fs)-3*cos(2*pi*204*t/Fs)+4*sin(2*pi*208*t/Fs);
f1=200;
k=0:1:5119;
h=exp((-j*2*pi*f1*k)/Fs);
y=x.*h;
y1=y(1:10:5120);
```

```
k1=0:1:511;
h1=exp((j*2*pi*f1*k1)/512);
y1=y1.*h1;
y1=abs(fft(y1));
m=0:511;
subplot(221);plot(m,y1);axis([200,220,0,1000]);
title('细化10倍后的频率特性');
t=0:1:4095;Fs=5120;
x=sin(2*pi*202*t/Fs)-3*cos(2*pi*204*t/Fs)+4*sin(2*pi*208*t/Fs);
f1=200;w1=2*pi*f1;
k=0:1:4095;
h=exp((-j*2*pi*f1*k)/Fs);
y=x.*h;
y1=y(1:8:4096);
k1=0:1:511;
h1=exp((j*2*pi*f1*k1)/512);
y1=y1.*h1;
y1=abs(fft(y1));
m=0:511;
subplot(222);plot(m,y1);axis([200,220,0,800]);
title('细化8倍后的频率特性');
t=0:1:2047;Fs=5120;
x=sin(2*pi*202*t/Fs)-3*cos(2*pi*204*t/Fs)+4*sin(2*pi*208*t/Fs);
f1=200;w1=2*pi*f1;
k=0:1:2047;
h=exp((-j*2*pi*f1*k)/Fs);
y=x.*h;
y1=y(1:4:2048);
k1=0:1:511;
h1=exp((j*2*pi*f1*k1)/512);
y1=y1.*h1;
y1=abs(fft(y1));
m=0:511;
subplot(223);plot(m,y1);axis([200,220,0,1000]);
title('细化4倍后的频率特性');
t=0:1:1023;Fs=5120;
x=sin(2*pi*202*t/Fs)-3*cos(2*pi*204*t/Fs)+4*sin(2*pi*208*t/Fs);
f1=200;w1=2*pi*f1;
k=0:1:1023;
h=exp((-j*2*pi*f1*k)/Fs);
y=x.*h;
y1=y(1:2:1024);
k1=0:1:511;
h1=exp((j*2*pi*f1*k1)/512);
y1=y1.*h1;
```

```
y1=abs(fft(y1));
m=0:511;
subplot(224);plot(m,y1);axis([200,220,0,1000]);
title('细化2倍后的频率特性');
figure;
t=0:1:511;Fs=5120;
x=sin(2*pi*202*t/Fs)-3*cos(2*pi*204*t/Fs)+4*sin(2*pi*208*t/Fs);
y1=abs(fft(x));
m=0:511;f=10*t;
subplot(111);plot(f,y1);axis([0,500,0,700]);
title('细化前频率特性');
```

3.1.9 用 DSP 进行频谱监控

在通信系统中，为了保证正常的通信，系统应能够动态地监测信道的质量。频谱监测系统可以使用 FFT 实现。对信道的时域采样信号进行 FFT 变换后，在频域中明显地对应了不同的信道。从 FFT 可以计算功率谱，它代表了特定信道在当前这个时间段的信号幅度，随着时间的推移不断进行 FFT 和功率谱的计算，就可以得到某一个特定信道随时间变化的信号强度，从而据此判断信道受到噪声干扰的强度。

程序 s343.m 是用 MATLAB 实现的频谱监控，它可以动态显示输入信号和信号的频谱，如图 3-36 所示。

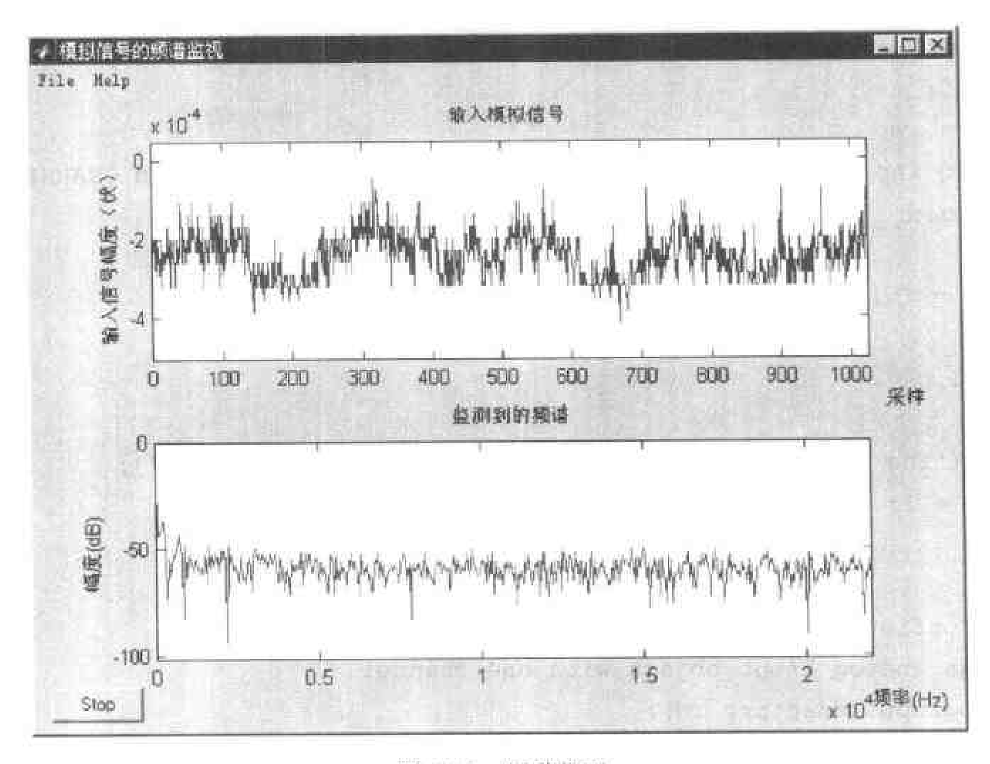


图 3-36 频谱监测

```
%s343.m
function demoft(varargin)
%DEMODF_FFT FFT display of an incoming analog input signal.
switch nargin
```

```
case 0
   data = localInitAI; % Create the analog input object.
   data = localInitFig(data); % Create the  figure.
   hFig = data.handle.figure;
case 1
   data = []; % Initialize variables.
   action=varargin{1};
   hFig=varargin{2};
   try
      data=get(hFig,'Userdata');
   end
  % Based on the action, call the appropriate local function.
  % Create the  figure.
      data = localInitFig(data);
      hFig = data.handle.figure;
   end
% Update the analog input object's UserData.
if isvalid(data.ai)
   set(data.ai, 'Userdata', data);
end
% Create the object and get the first fft.
function [data, errflag] = localInitAI(varargin)
% Initialize variables.
errflag = 0;
data = [];
% Either no input arguments or all three - ADAPTORNAME, ID and CHANNELID.
switch nargin
case 0
   adaptor = 'winsound';
   id = 0;
   chan = 1;
otherwise
   lasterr('The ADAPTORNAME, ID and CHANID must be specified.');
   errflag = 1;
   return;
end
% Object Configuration.
% Create an analog input object with one channel.
ai = analoginput(adaptor, id);
addchannel(ai, chan);
% Configure the analog input object.
set(ai, 'SampleRate', 44100);
% Configure the callback to update the display.
set(ai, 'TimerAction', {'demoai_fft', 'fftsbowdata'});
set(ai, 'TimerPeriod', 0.1);
```

```
% Configure the analog input object to trigger manually twice.
set(ai, 'SamplesPerTrigger', 1024);
set(ai, 'TriggerRepeat', 1);
set(ai, 'TriggerType', 'manual');
% Object Execution.
% Start the analog input object.
start(ai);
trigger(ai);
% Obtain the available time and data.
[d,time] = getdata(ai, ai.SamplesPerTrigger);
% Calculate the fft.
Fs = get(ai, 'SampleRate');
blockSize = get(ai, 'SamplesPerTrigger');
[f,mag] = localDaqfft(d,Fs,blockSize);
% Create the data structure.
data.ai = ai;
data.getdata = [d time];
data.daqfft = [f mag];
data.handle = [];
% Set the object's UserData to data.
set(data.ai, 'Userdata', data);
% Create the display.
function data = localInitFig(data)
% Initialize variables.
btnColor=get(0,'DefaultUIControlBackgroundColor');
% Position the GUI in the middle of the screen
screenUnits=get(0,'Units');
set(0,'Units','pixels');
screenSize=get(0,'ScreenSize');
set(0,'Units',screenUnits);
figWidth=600;
figHeight=360;
figPos=[(screenSize(3)-figWidth)/2(screenSize(4)-figHeight)/2figWidth
figHeight];
% Create the figure window.
hFig=figure(...
   'Color'             ,btnColor              ,...
   'IntegerHandle'     ,'off'                 ,...
   'DoubleBuffer'      ,'on'                  ,...
   'DeleteFcn'         ,'demoai_fft(''close'',gcbf)',...
   'MenuBar'           ,'none'                ,...
   'HandleVisibility'  ,'on'                  ,...
   'Name'              ,'模拟信号的频谱监视'   ,...
   'Tag'               ,'模拟信号的频谱监视'   ,...
   'NumberTitle'       ,'off'                 ,...
```

```
    'Units'             ,'pixels'              ,...
    'Position'          ,figPos                ,...
    'UserData'          ,[]                    ,...
    'Colormap'          ,[]                    ,...
    'Pointer'           ,'arrow'               ,...
    'Visible'           ,'on');
% Create Data subplot.
hAxes(1) = axes(...
    'Position'          , [0.1300 0.5811 0.7750 0.3439],...
    'Parent'            , hFig,...
    'XLim'              , [0 get(data.ai, 'SamplesPerTrigger')],...
    'YLim'              , [-0.5 0.5]);
% Plot the data.
hLine(1) = plot(data.getdata(:,1));
set(hAxes(1), 'XLim', [0 get(data.ai, 'SamplesPerTrigger')]);
% Label the plot.
xlabel('采样');
ylabel('输入信号幅度 (V) ');
title('输入模拟信号');
% Create the FFT subplot.
hAxes(2) = axes(...
    'Position'          , [0.1300 0.1100 0.7750 0.3439],...
    'Parent'            , hFig,...
    'XLim'              , [0 max(data.daqfft(:,1))]...
    );
% Plot the data.
hLine(2) = plot(data.daqfft(:,1),data.daqfft(:,2));
set(hAxes(2), 'XLim', [0 max(data.daqfft(:,1))]);
% Label the plot.
xlabel('频率(Hz)');
ylabel('幅度(dB)');
title('监测到的频谱');
% Create a start/stop pushbutton.
htoggle = uicontrol(...
    'Parent'          , hFig,...
    'Style'           , 'pushbutton',...
    'Units'           , 'normalized',...
    'Position'        , [0.0150 0.0111 0.1 0.0556],...
    'Value'           , 1,...
    'String'          , 'Stop',...
    'Callback'        , 'demoai_fft(''stop'', gcbf);');
hmenu(1) = uimenu('Parent', hFig,'Label', 'File');
hmenu(2) = uimenu(hmenu(1),'Label', 'Close demoai_fft',...
    'Callback', 'demoai_fft(''close'',gcbf)');
hmenu(3) = uimenu('Parent', hFig,'Label', 'Help');
```

```
hmenu(4) = uimenu(hmenu(3),'Label', 'Data Acquisition Toolbox',...
   'Callback', 'helpwin(''daq'')');
hmenu(5) = uimenu(hmenu(3),'Label', 'demoai_fft',...
   'Callback', 'helpwin(''demoai_fft'')');
% Store the handles in the data matrix.
data.handle.figure = hFig;
data.handle.axes = hAxes;
data.handle.line = hLine;
data.handle.toggle = htoggle;
data.state = 0;
% Set the axes handlevisibility to off.
set(hAxes, 'HandleVisibility', 'off');
% Store the data matrix and display figure.
set(hFig,'Visible','on','UserData',data,'HandleVisibility', 'off');
% Close the figure window.
function localClose(data)
% Stop the device if it is running and delete the object.
if isvalid(data.ai)
   if strcmp(get(data.ai, 'Running'), 'On')
      stop(data.ai);
   end
   delete(data.ai);
end
% Close the figure window.
delete(data.handle.figure);
% Stop or start the device.
function data = localStop(data)
% Based on the state either stop or start.
if data.state == 0
   stop(data.ai); % Stop the device.
   set(data.handle.toggle, 'String', 'start');
   data.state = 1; % Store the new state.
else
   start(data.ai); % Start the device.
   set(data.handle.toggle, 'String', 'stop');
   data.state = 0; % Store the new state.
end
% 计算数据的 FFT
function [f, mag] = localDaqfft(data,Fs,blockSize)
% Calculate the fft of the data.
xFFT = fft(data);
xfft = abs(xFFT);
% Avoid taking the log of 0.
index = find(xfft == 0);
xfft(index) = 1e-17;
```

```
mag = 20*log10(xfft);
mag = mag(1:blockSize/2);
f = (0:length(mag)-1)*Fs/blockSize;
f = f(:);
function localfftShowData(obj,event)
% Get the handles.
data = obj.UserData;
hFig = data.handle.figure;
hAxes = data.handle.axes;
hLine = data.handle.line;
% Execute a peekdata.
x = peekdata(obj, obj.SamplesPerTrigger);
% 计算 FFT
Fs = obj.SampleRate;
blockSize = obj.SamplesPerTrigger;
[f,mag] = localDaqfft(x,Fs,blockSize);
% 更新曲线
set(hLine(1), 'YData', x(:,1));
set(hLine(2), 'XData', f(:,1), 'YData', mag(:,1));
drawnow;
```

3.2 二维信号处理的应用

数字图像处理作为一门学科可以追溯到20世纪60年代初期。1964年美国喷气推进实验室对太空飞船发回的月球照片处理后，使原本模糊不清的图像变得清晰逼真，收到了令人满意的效果，标志着数字图像这门学科的诞生。几十年来，数字图像处理技术向着更深入、更广泛的方向发展，在军事、气象、医学、通讯等领域大显身手，成为现代高技术中一个生机勃勃的研究方向。

一般来说，图像处理的目的在于：

① 提高图像质量，使模糊的图像变得清晰；

② 提取图像的有效特征，以便进行模式识别；

③ 通过图像变换和有效编码来压缩其频带或数据，以便传输和存储。

3.2.1 计算机数字图像

计算机数字图像文件常用格式：PCX、BMP、JPEG、TIFF、HDF等，MATLAB中的数字图像是由一个或多个矩阵表示的。在缺省的情况下，MATLAB将图像中的数据存储为双精度（double），即64位的浮点数，这种存储方法的优点在于使用中不需要数据类型转换。然而对于图像存储来说，用64bit来表示图像数据会导致巨大的存储量，所以MATLAB还支持另一种类型无符号整型（unit8），即图像矩阵中的每个数精占用一个字节。由于MATLAB的大多数操作和运算都不支持unit8类型，在涉及运算时要将其转换成double型。

3.2.1.1 数字图像的类型

MATLAB图像处理工具箱支持四种基本图像类型：真彩色（RGB）图像、索引图像、

灰度图像、二进制图像。

（1）真彩色图像

真彩色图像用 R、G、B 三个分像表示一个像素的颜色，所以对一个尺寸为 $m\times n$ 的真彩色图像来说，其数据结构就是一个 $m\times n\times 3$ 的多维数组。若要读取图像中（6,5）处的像素像，可以查看三元组（6,5,1:3）。

真彩色图像可用 double 型来存储，此时亮度像范围是[0,1]。

[例 1] 真彩色图像“s1.bmp”如图 3-37，显示它的结构和大小。

图 3-37　真彩色图像示例

```
%s336.m
R=imread('s1.bmp');
whos R                    %查看 R 的大小、类型
B=double(R)/255           % uint8 到 double 的转换
```

运行结果如下。

Name	Size	Bytes	Class
R	6×8×3	144	uint8 array

Grand total is 144 elemeuts using 144 bytes

图像 s1.bmp 的 RGB 图像结构见表 3-1。

表 3-1　真彩色图像的结构

	1	1	0	0	0	1	0	1
	1	1	0	0	0	1	0	1
红	1	1	0	0	0	1	0	1
B(:,:,1)	1	1	0	0	0	1	0	1
	1	1	0	0	0	1	0	1
	1	1	0	0	0	1	0	1
	0	1	1	1	0	0	0	1
	0	1	1	1	0	0	0	1
绿	0	1	1	1	0	0	0	1
B(:,:,2)	0	1	1	1	0	0	0	1
	0	1	1	1	0	0	0	1
	0	1	1	1	0	0	0	1
	0	0	0	1	1	1	0	1
	0	0	0	1	1	1	0	1
蓝	0	0	0	1	1	1	0	1
B(:,:,3)	0	0	0	1	1	1	0	1
	0	0	0	1	1	1	0	1
	0	0	0	1	1	1	0	1

（2）索引图像

索引图像包含两个结构，一个是调色板，一个是图像数据矩阵。调色板是一个有三列和若干行的色彩映像矩阵，矩阵的每一行代表一种颜色，通过三个分别代表红、绿、蓝颜色的双精度数，形成一种特定的颜色。

MATLAB 中调色板的色彩强度是[0,1]中的浮点数，0 代表最暗，1 代表最亮。表 3-2 是常用颜色的 R、G、B 值。

表 3-2　常用颜色的 R、G、B 值

颜色	黑	白	红	绿	蓝	黄	洋红	青蓝	天蓝	橘黄	深红	灰
R	0	1	1	0	0	1	1	0	0.67	1	0.5	0.5
G	0	1	0	1	0	1	0	1	0	0.5	0	0.5
B	0	1	0	0	1	0	1	1	1	0	0	0.5

MATLAB 亮供了 10 个函数用于产生预存的标准调色板，见表 3-3。

表 3-3　产生标准调色板的函数

函数名	调　色　板	函数名	调　色　板
Hsv	色彩饱和度，以红色开始并以红色结束	Bone	带蓝色的灰度
Hot	黑色-红色-黄色-白色	Jet	Hsv 的一种变形，以蓝色开始并以蓝色结束
Cool	青蓝和洋红的色度	Copper	线性铜色度
Pink	粉红的色度	Prim	三棱镜，交替为红、橘黄、黄、绿和天蓝
Gray	线性灰度	Flag	交替为红、白、蓝和黑

索引图像的数据类型为 double 和 unit8。当图像数据为 double 类型时，值 1 代表调色板中的第 1 行，值 2 代表调色板中的第 2 行；当图像数据为 unit8 亮型时，值 0 代表调色表中的第 1 行，值 1 代表调色板中的第 2 行。

（3）灰度图像

灰度图像只需要一个数据矩阵，数据亮型可以是 double，值的范围为[0,1]；也可以是 unit8，值的范围为[0,255]。

（4）二进制图像

二进制图像也只需要一个数据矩阵，每个像素只取两个灰度值。可以采用 double 或 unit8 类型存储。

判断图像类型函数如下。

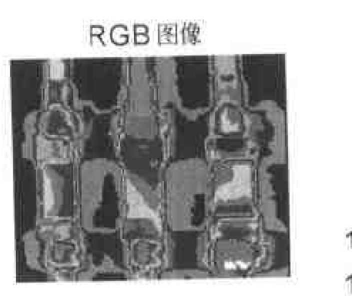
flag=isbw(X)，如果图像 X 为黑白图像返回 1，否则返回 0。

flag=isrgb(X)，如果图像 X 为真彩色图像返回 1，否则返回 0。

flag=isind(X)，如果图像 X 为索引图像返回 1，否则返回 0。判断 X 是否为索引图储的准则是：如像 X 是双表度类，其所有元素必须是大于或等于 1 的整数，并且 X 的储数必须为 2；如果 X 是 unit8 类，其逻辑标志必须是关闭的，并且 X 的储数必须为 2。

flag=isgray(X)，如果图像 X 为灰度图像返回 1，否则返回 0。判断 X 是否为灰度图像的准则是：如果 X 是双特度类，其所有元素必须在[0,1]范围内，并且 X 的维数必须为 2；如像 X 是 unit8 亮，X 的维数必须为 2。

[例 2]　判断图像“4pf1.tif”和“tartan”的亮型，并显示图像，结像如图 3-38 所示。



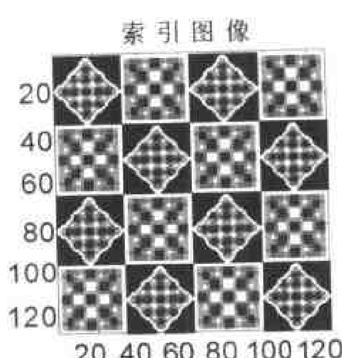


图 3-38　图像的显示

```
%s331.m
I1=imread('4pf1.tif');
load tartan;
f1=isbw(I1),f2=isrgb(I1),f3=isind(I1),f4=isgray(I1)
f5=isbw(X),f6=isrgb(X),f7=isind(X),f8=isgray(X)
subplot(121);imshow(I1);
subplot(122);subimage(X,map);
```

结果 f1=0　f2=1　f3=0　f4=0　f5=0　f6=0　f7=1　f8=0

说明图像“4pf1.tif”是真彩色图像，图像“tartan”是索引图像。

3.2.1.2　图像类型的转换

许多图像处理工作都要求对图像类型有特定的要求。比如要对一幅索引色图像滤波，必须把它转换成真彩色图像，否则结果是毫无意义的。

（1）灰度图像与索引色图像的互换

[X,MAP]=gray2ind(I,n)，将灰度图像 I 转换成具有颜色图 MAP 的索引图像 X，n 缺省值为 64。

I=ind2gray(X,MAP)，将具有颜色图 MAP 的索引图 X 转换成为灰度图像 I，X 可以是 unit8 或双精度类型，I 为双精度类型。

（2）RGB 图像与索引色图像的互换

[X,MAP]=rgb2ind(R)，直接将 RGB 图像 R 转换成具有颜色图 MAP 的矩阵 X。由于每个像素点具有一个值，转换后的颜色图可能很长。

[X,MAP]=rgb2ind(R,tol)，用均匀量化的方法，将 RGB 图像 R 转换成索引图像 X。MAP 包括至少(Floor(1/tol)+1)^3 个颜色。tol 必须从 0.0 至 1.0。

[X,MAP]=rgb2ind(R,n)，使用最小方差量化方法，将 RGB 图像 R 转换成索引图像 X。MAP 中包括 n 个颜色。

X=rgb2ind(R,MAP)，通过匹配 RGB 中的颜色与颜色图 MAP 中最相近的颜色，将 RGB 图像 R 转换为具有 MAP 颜色图的索引图像。

R=ind2rgb(X,MAP)，将矩阵 X（unit8 或双精度格式）及其相应的颜色图 MAP 转换成 RGB 真彩色格式（m×n×3 双精度数组）图像 R。

[例 3]　将真彩色图像“4pf1.tif”转换成索引图像，结果如图 3-39 所示。

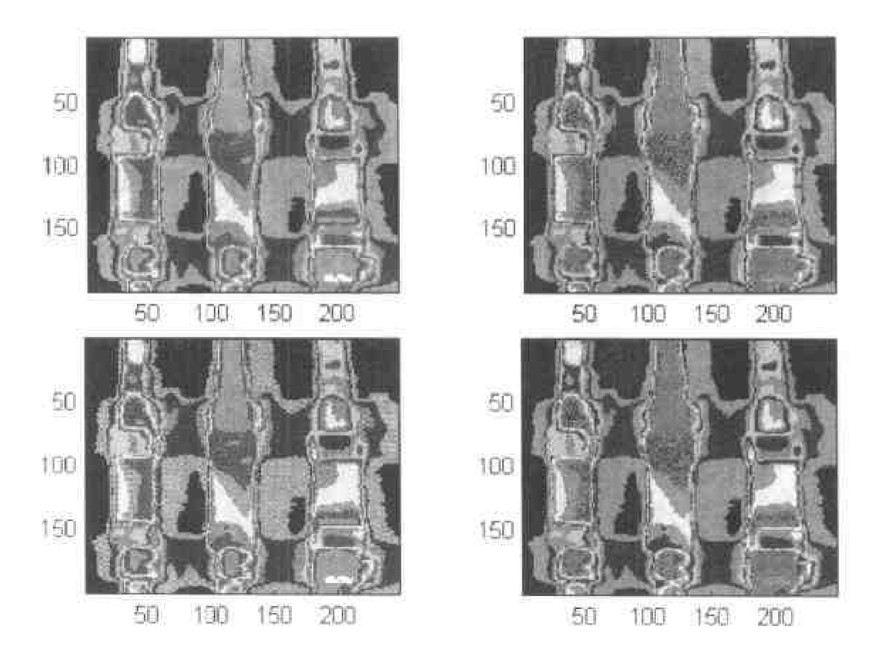


图 3-39 RGB图像转换成索引图像

```
%s332.m
R=imread('4pf1.tif');
[X1,MAP1]=rgb2ind(R);
[X2,MAP2]=rgb2ind(R,8);
[X3,MAP3]=rgb2ind(R,0.2);
X4=rgb2ind(R,MAP2);
subplot(221);subimage(X1,MAP1);
subplot(222);subimage(X2,MAP2);
subplot(223);subimage(X3,MAP3);
subplot(224);subimage(X4,MAP2);
```

[例4] 将索引图像“tartan”转换为真彩色图像，结果如图3-40所示。

```
%s333.m
load tartan;
R=ind2rgb(X,Hsv);
subplot(121);subimage(X,map);
subplot(122);subimage(R);
```

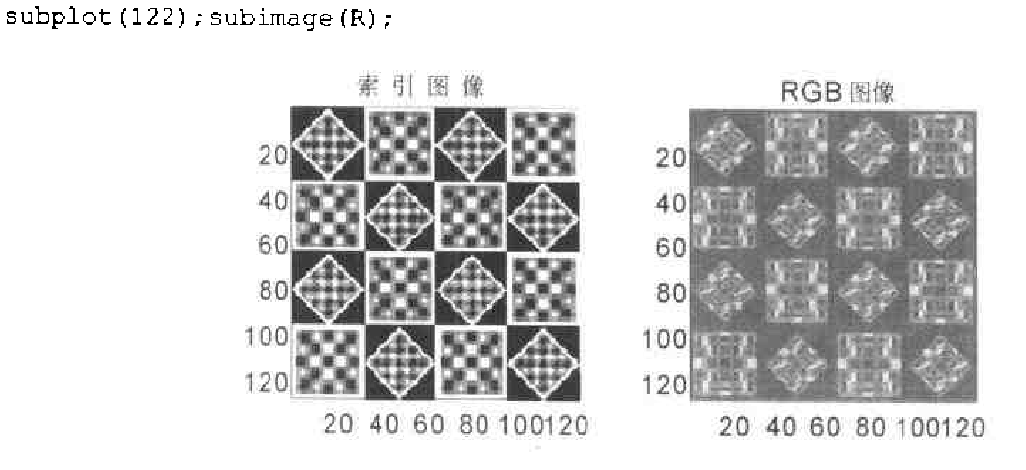


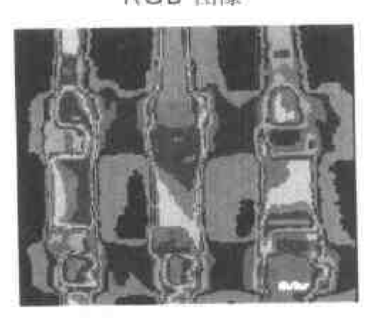
图 3-40 索引图像转换为RGB图像

（3）RGB 图像转换为灰度图像

I=rgb2gray(R)，将真彩色图像 R 转换为灰度图像 I。

[例 5] 将真彩色图像“4pf1.tif”转换为灰度图像，结果如图 3-41 所示。

```
%s334.m
R=imread('4pf1.tif');
I=rgb2gray(R);
subplot(121);imshow(R);
subplot(122);imshow(I);
```



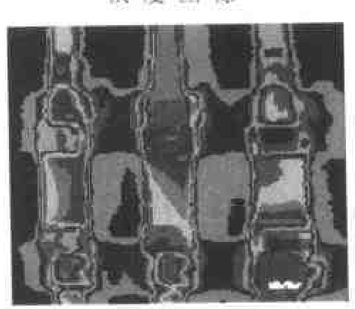


图 3-41 RGB 图像转换为灰度图像

（4）将图像转换为二进制图像

使用 im2bw 可以将索引图像、灰度图像和 RGB 图像转换为二进制（黑白）图像。在转换过程中，如果输入图像不是灰度级图像，首先将其转换为灰度级别图像，然后通过阈值化将灰度级图像转换成为二进制图像。输出图像在输入图像所有亮度小于给定值(level)像素点处均为 0，在其他地方均为 1。

BW=im2bw(I,level)，转换灰度图像 I 为黑白图像。

BW=im2bw(X,MAP,level)，将带有颜色图 MAP 的索引图像 X 转换为黑白图像。

BW=im2bw(R,level)，转换 RGB 图像 R 为黑白图像。

[例6] 将灰度图像“rice.tif”、RGB 图像“4pf1.tif”和索引图像“tartan”转换为二进制（黑白）图像。结果如图 3-42 所示。

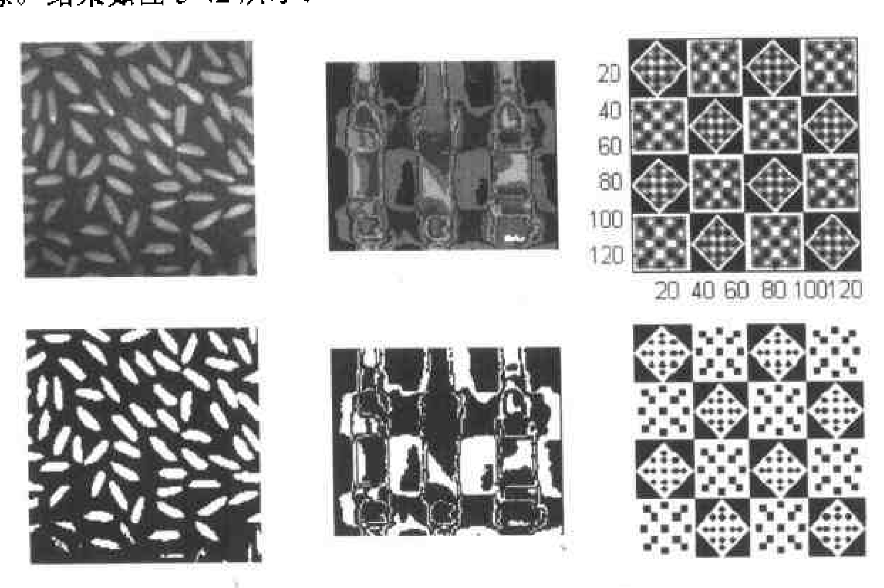


图 3-42 图像转换为二进制图像

```
%s335.m
I=imread('rice.tif');
R=imread('4pf1.tif');
load tartan;
subplot(231);imshow(I);
subplot(232);imshow(R);
subplot(233);subimage(X,map);
BW1=im2bw(I,0.5);
BW2=im2bw(R,0.5);
BW3=im2bw(X,Hsv,0.5);
subplot(234);imshow(BW1);
subplot(235);imshow(BW2);
subplot(236);imshow(BW3);
```

3.2.1.3 数字图像文件的读/写和显示

（1）图像文件的读/写

图像文件的读写可以用函数 imread，用它读入*.bmp、*.jpg、*.pcx、*.tif、*.xwd、*.hdf 等格式图像文件。

用 imwrite 将图像文件写入到*.bmp、*.jpg、*.pcx、*.tif、*.xwd、*.hdf 等格式的图像文件中。

对于读入内存的图像，可以用 save 命令将其以二进制数据形式保存到以 mat 为扩展名的文件中，以后就可以用 load 命令将其调入到内存。

用 imfinfo 查看图像文件的信息。

[例 7] 读写图像，并显示信息。

```
%s337.m
R=imread('4pf1.tif'); %读入图像文件
imwrite(R,'4pf11.tif','tif'); %将图像输出到文件
save 4pf1 R; %将读入内存的图像以二进制数据形式保存
load 4pf1; %调入二进制数据文件
whos R
info1=imfinfo('4pf1.tif') %显示图像信息
info2=imfinfo('4pf11.tif')
```

结果如下。

```
Name        Size            Bytes    Class
R           199×248×3       148056   uint8 array
Grand total is 148056 elements using 148056 bytes
info1 =
                  Filename: '4pf1.tif'
               FileModDate: '25-Dec-1996 10:33:12'
                  FileSize: 148272
                    Format: 'tif'
             FormatVersion: []
```

```
                  Width: 248
                 Height: 199
               BitDepth: 24
              ColorType: 'truecolor'
        FormatSignature: [73 73 42 0]
              ByteOrder: 'little-endian'
         NewSubfileType: 0
          BitsPerSample: [8 8 8]
            Compression: 'Uncompressed'
PhotometricInterpretation: 'RGB'
info2=
               Filename: '4pf11.tif'
            FileModDate: '19-Jul-2001 11:36:28'
               FileSize: 92852
                 Format: 'tif'
          FormatVersion: []
                  Width: 248
                 Height: 199
               BitDepth: 24
              ColorType: 'truecolor'
        FormatSignature: [73 73 42 0]
              ByteOrder: 'little-endian'
         NewSubfileType: 0
          BitsPerSample: [8 8 8]
            Compression: 'PackBits'
PhotometricInterpretation: 'RGB'
```

（2）图像文件的显示

① 图像显示

imshow(I,n)，使用 n 个灰度级显示灰度图像，n 缺省时使用 256 级灰度或 64 级灰度显示图像。

imshow(I,[LOW,HIGH])，将 I 显示为灰度图像，并指定灰度级为范围[LOW,HIGH]。

imshow(X,ma)，使用颜色图 map 显示索引图像。

imshow(R)，显示真彩色图像。

imshow(FILEMANE)，显示 FILEMANE 所指定的图像文件。

② 在一个图形窗口中显示多幅图像

subimage 函数在一个图形窗口内使用多个调色板，与 subplot 联合使用，使得各种图像能在一个图形窗口内显示。与 subplot 的区别是：subplot 在图形窗口内的调色板只能是一个。

[例 8] 同时显示具有不同调色板的图像，结果如图 3-43 所示。

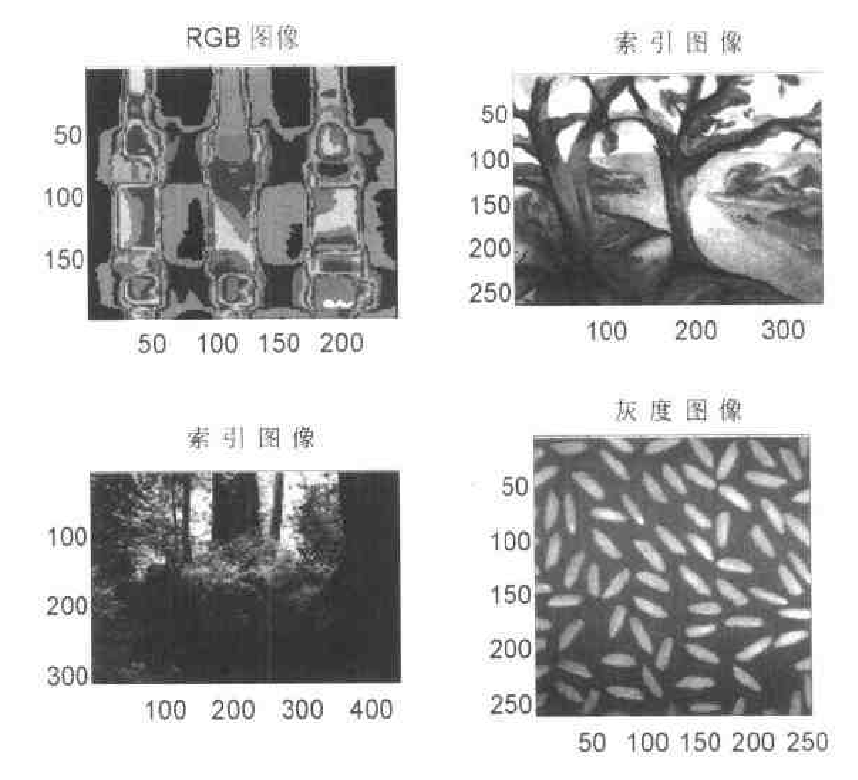


图 3-43 在同一窗口中显示不同调色板的图像

```
%s338.m
I=imread('4pf1.tif');
load trees
[I1,map1]=imread('forest.tif');
I2=imread('rice.tif');
subplot(221);subimage(I); %显示真彩色图像
subplot(222);subimage(X,map); %显示具有调色板 map 的图像
subplot(223);subimage(I1,map1); %显示具有调色板 map1 的图像
subplot(224);subimage(I2); %显示灰度图像
```

3.2.2 图像的消噪处理

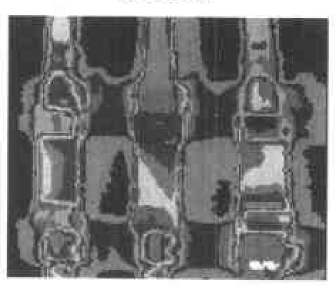
实际中摄取的图像一般都含有某种噪声，引起噪声的原因很多，噪声的种类也很多，可以将噪声分为加性噪声和乘性噪声。加性噪声中包含高斯噪声、椒盐噪声等典型噪声。MATLAB 图像处理工具箱提供了模拟噪声生成的函数 imnoise，可以对图像添加一些典型的噪声。

[例 9] 对用热像仪摄取的三相保险热图像“4pf1.tif”分别添加高斯噪声、椒盐噪声和乘性噪声。

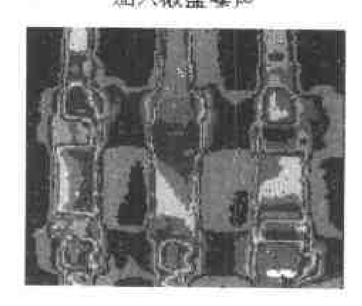
```
%s320.m
X=imread('4pf1.tif');
subplot(221);imshow(X);title('原始图像');
Y1=imnoise(X,'gaussian',0.02);
Y2=imnoise(X,'salt&pepper',0.02);
Y3=imnoise(X,'speckle',0.03);
subplot(222);imshow(Y1);title('加入高斯噪声');
```

```
subplot(223);imshow(Y2);title('加入椒盐噪声');
subplot(224);imshow(Y3);title('加入乘性噪声');
```

结果如图 3-44 所示。





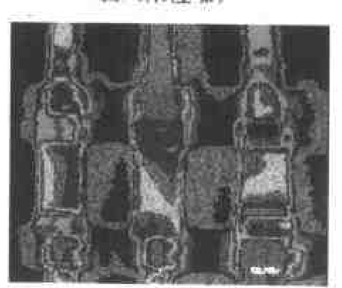


图 3-44 加入典型噪声的图像

图像消噪的方法很多，主要采用中值滤波、顺序统计滤波和自适应除噪滤波技术。

3.2.2.1 中值滤波

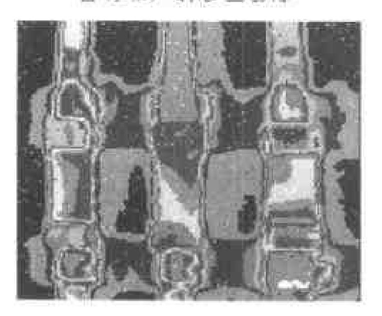
直接在空间域上对图像进行平滑处理。假设图像上的噪声是加性的、互不相关的，且均值为零。一幅有噪声的数字图像包括：原始信号 $f(x,y)$ 和噪声 $n(x,y)$，即：$g(x,y)=f(x,y)+n(x,y)$，经过处理后得到的平滑图像为

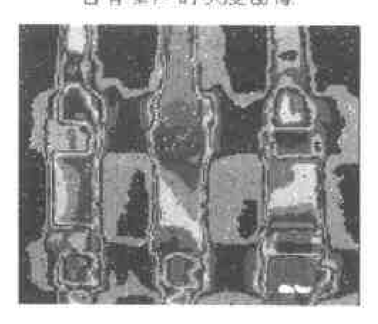
$$g(x,y)=\frac{1}{M}\sum_{(i,j)\in s}g(i,j)=\frac{1}{M}\sum_{(i,j)\in s}f(i,j)+\frac{1}{M}\sum_{(i,j)\in s}n(i,j)$$

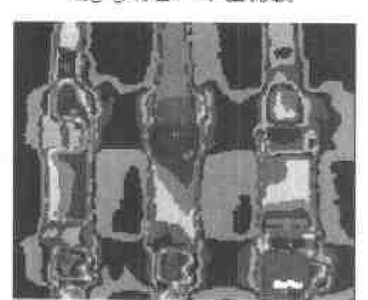
式中 s 是点（x,y）邻域内的点集，M 是点集 s 中的总数。图像经过平滑处理后，可使噪声方差减小 M 倍。但简单的局部平均使图像模糊，特别是黑白交界处轮廓变得不清晰。为此采用中值滤波法：就是取像素点邻域中各像素灰度的中值作为滤波器的输出。

[例 10] 对加入椒盐噪声的三相保险热图像“4pfl.tif”作中值滤波，结果如图 3-45 所示。

```
%s321.m
X=imread('4pfl.tif');
J=imnoise(X,'salt&pepper',0.02);
subplot(221);imshow(J);title('含有噪声的彩色图像');
J1=rgb2gray(J); %将彩色图像转换为灰度图像
K1=medfilt2(J1);K2=medfilt2(J1,[5,4]);
subplot(222);imshow(J1);title('含有噪声的灰度图像');
subplot(223);imshow(K1);title('用 3*3 的窗口中值滤波');
subplot(224);imshow(K2);title('用 5*4 的窗口中值滤波');
```







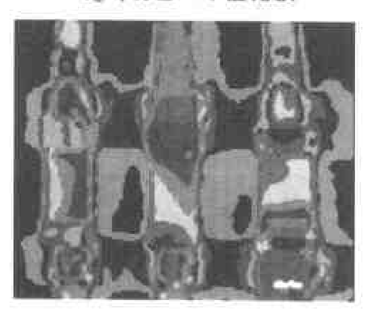


图3-45 图像的中值滤波

3.2.2.2 自适应除噪滤波

估计每个像素的局部均值与方差：

$$\mu = \frac{1}{MN}\sum_{(n_1,n_2)\in\eta} a(n_1,n_2)$$

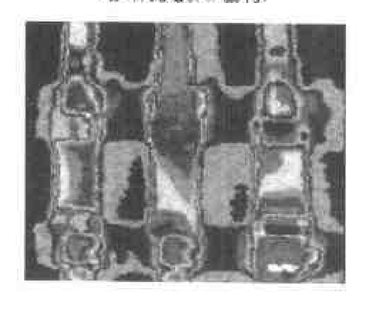
$$\sigma^2 = \frac{1}{MN}\sum_{(n_1,n_2)\in\eta} a^2(n_1,n_2) - M^2$$

其中η为图像中每个像素的局部邻域，维纳滤波估计为

$$b(n_1,n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2}[a(n_1,n_2) - \mu]$$

其中v为噪声方差，如果没有给出，则自动以所有局部估计方差的均值代替。

[例11] 对加入高斯噪声的三相保险热图像“4pf1.tif”作维纳自适应滤波，结果如图3-46所示。



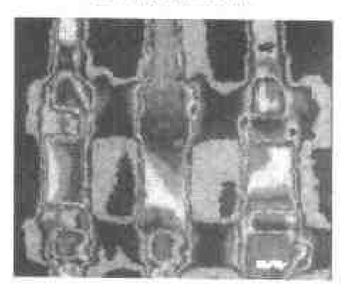


图3-46 维纳滤波后图像

```
%s322.m
X=imread('4pfl.tif');
Y=rgb2gray(X);
J=imnoise(Y,'gaussian',0.02);
K1=wiener2(J,[3,4]); %使用 3×4 邻域
K2=wiener2(J,[5,4]); %使用 5×4 邻域
subplot(121);imshow(K1);title('维纳滤波后图像');
subplot(122);imshow(K2);title('维纳滤波后图像');
```

3.2.3 图像对比度增强

实际中摄取到的图像往往会出现失真，虽然在多数情况下难以对图像降质的具体过程建立精确的数学模型，但可以大致估计出一些引起降质的原因，从而采取一些简单易行的方法改善图像的质量。

3.2.3.1 对比度增强

由于曝光不足等因素的影响，图像的亮度分布会发生一些非线性失真，常常表现为对比度不强、图像的整体感觉较暗等等。通过对比度调整可以改善这种失真。

（1）线性变换

设原图像 $f(x,y)$ 的灰度范围为$[m,n]$，所希望的图像灰度分布为$[M,N]$，则对比度调整的线性变换为

$$g(x,y)=\frac{N-M}{n-m}[f(x,y)-m]+M$$

通过对系数 $\frac{N-M}{n-m}$ 的设计，就可以使直方图伸展或压缩，并使整体亮度趋于亮区或暗区。图 3-47 是直方图线性变换的关系曲线。

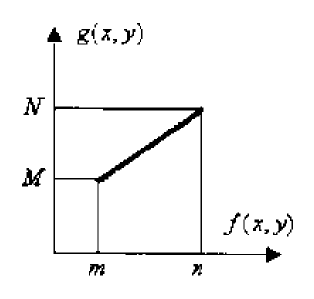


图 3-47 直方图线性变换

（2）非线性变换（γ 矫正）

对直方图的变换可通过在线性变换系数上乘矫正量 γ 实现非线性变换。图 3-48 是 γ 取不同值时的非线性变换。

（3）灰度倒置变换

变换公式为：

$$g(x,y)=g_M-(g_M/f_M)f(x,y)$$

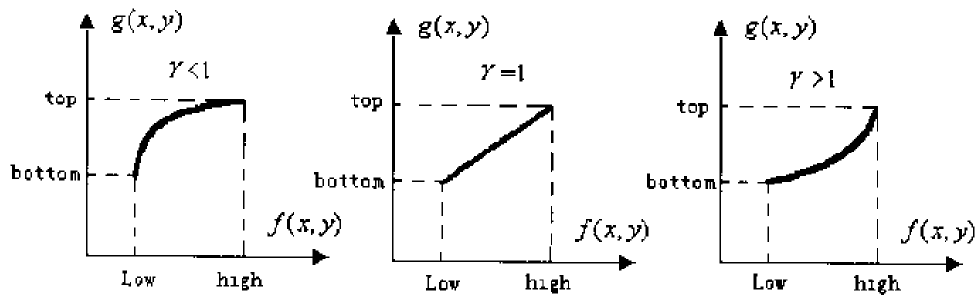


图 3-48 直方图非线性变换

函数imadjust实现上述对比度调整算法，格式为：

J=imadjust(I,[low high],[bottom top],gamma)

将灰度图像I转换为图像J，使值从low到high与从bottom到top相匹配。值大于high或小于low的值被剪去。使用该函数时可将[low high]或[bottom top]制定为空矩阵[]，此时缺省值为[0 1]。gamma用来指定描述I和J值关系曲线的形状；gamma<1，越亮输出值越加强；gamma>1，越亮输出值越减弱；gamma=1，表示线性变换。

[例12] 调整图像“pout.tif”的对比度，调整前后的图像、直方图如图3-49所示。

```
%s323.m
I=imread('pout.tif');
J=imadjust(I,[0.3 0.7 ],[]); %对比度调整
subplot(221);imshow(I);title('调整前图像');
subplot(222);imshow(J);title('调整后图像');
subplot(223);imhist(I);title('调整前图像的直方图');
subplot(224);imhist(J);title('调整后图像的直方图');
```

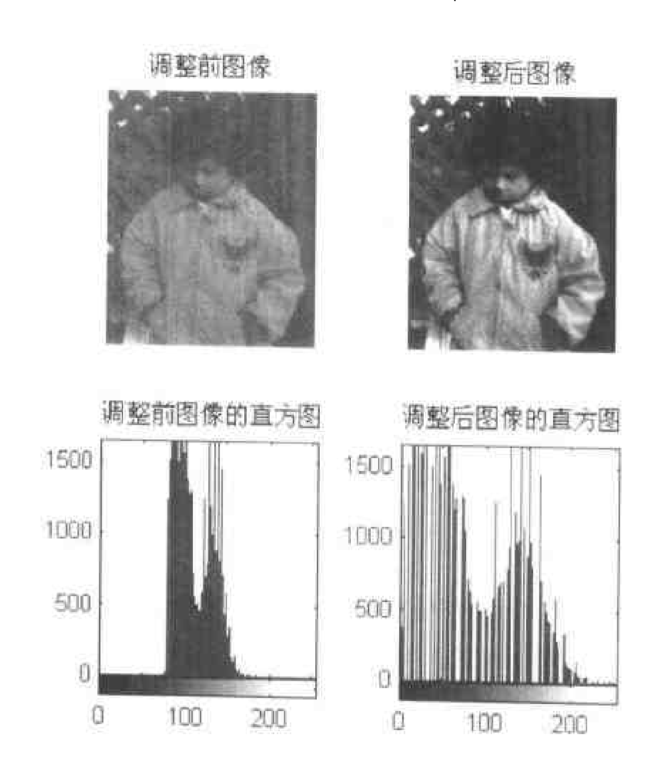


图3-49 对比度调整前后的图像、直方图

[例13] 调整图像“pout.tif”的对比度，并将调整后的图像的灰度倒置如图3-50所示。

```
%s324.m
I=imread('pout.tif');
J1=imadjust(I,[0.3 0.7 ],[0.1 0.9],0.9); %对比度调整
J2=imadjust(J1,[0 1],[1 0],0.9); %灰度倒置
subplot(221);imshow(J1);title('灰度图像');
subplot(222);imshow(J2);title('灰度倒置图像');
subplot(223);imhist(J1);title('灰度图像直方图');
subplot(224);imhist(J2);title('倒置图像直方图');
```

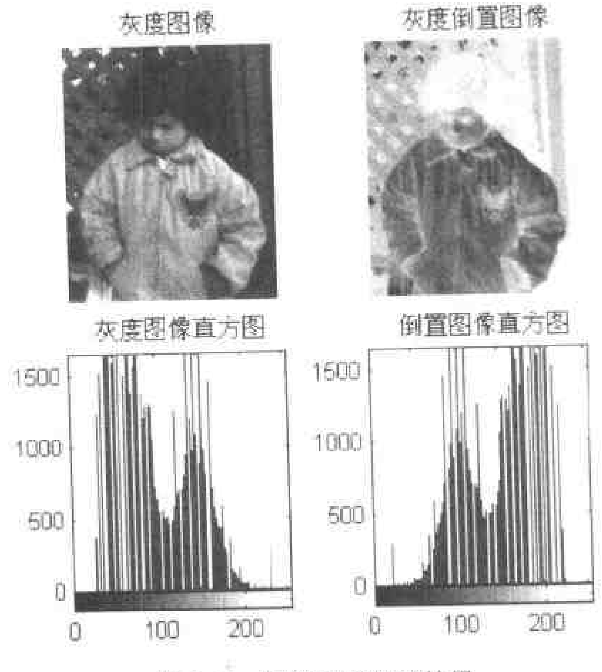


图 3-50　图像灰度倒置结果

3.2.3.2　直方图均衡化

均匀量化的自然图像的灰度直方图通常在低值灰度区间上频率较大，使得图像中较暗区域中的细节常常看不清楚。为了使图像清晰，可将图像的灰度范围拉开，并且让灰度频率较小的灰度级变大，即让灰度直方图在较大的动态范围内趋于一致。

设$\varphi_r(r)$为原始图像的灰度密度函数，则直方图均衡化后的图像为

$$g(x,y)=\int_0^r\varphi_r(u)\mathrm{d}u$$

对于数字图像，可以对上述公式做离散近似。设原图像的像素总数为N，灰度级的个数为L，第k个灰度级出现的频数为n_k。若原图像$f(x,y)$在像素点(x,y)处的灰度为r_x，则直方图均衡化后的图像$g(x,y)$在(x,y)处的灰度s_k为

$$s_k=\sum_{l=0}^{k}\frac{n_l}{N}$$

函数 histeq 用于直方图均衡增强对比度。

J–histeq(I,hgarm)将灰度图像 I 的直方图变成用户指定的向量 hgarm，hgarm 中的每一个元素都在[0，1]中。

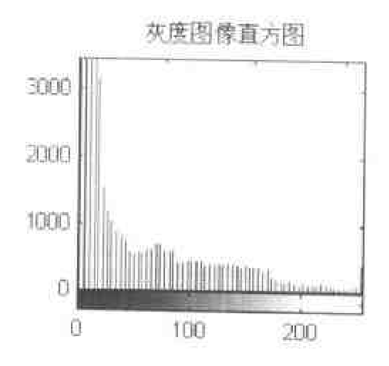
J=histeq(I,N) 将灰度图像 I 转换成具有 N 个离散灰度级的灰度图像 J，N 缺省值为 64。

[例 14]　对图像“tire.tif”作直方图均衡化，结果如图 3-51 所示。

```
%s325.m
I=imread('tire.tif');
J=histeq(I); %直方图均衡化
subplot(221);imshow(I);title('灰度图像');
subplot(222);imshow(J);title('直方图均衡化图像');
subplot(223);imhist(I,64);title('灰度图像直方图');
subplot(224);imhist(J,64);title('均衡化图像直方图');
```







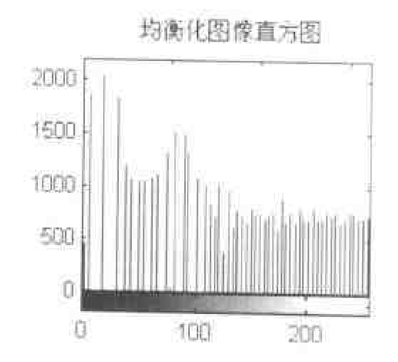
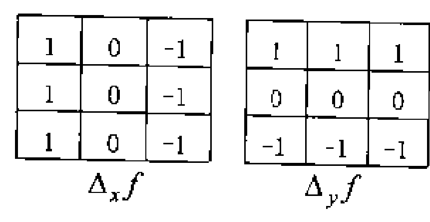


图 3-51　图像的直方图均衡化

3.2.4　图像的边缘检测

代表图像二维特征的是图像边缘和区域，从图像中提取这些特征有许多方法。边缘检测在图像处理与计算机视觉中占有特殊的位置，它是底层视觉处理中最重要的环节之一。



1	0	-1
1	0	-1
1	0	-1

$\Delta_x f$

1	1	1
0	0	0
-1	-1	-1

$\Delta_y f$

图 3-52　Prewitt 算子

边缘检测的基本思想是，首先利用边缘增强算子，突出图像中的局部边缘，然后定义像素的“边缘强度”，通过设置门限的方法提取边缘点集。常用的边缘检测算子有 Robert 算子、Sobel 算子、Prewitt 算子、LOG 算子和 Canny 算子。图 3-52 是 Prewitt 算子的模板。

函数 edge 用于检测灰度图像的边缘。

BW=edge(I,method)返回与 I 大小一样的二进制 BW，其中元素 1 为发现 I 中的边缘。Method 为下列字符串之一：

'sobel'　　缺省值，用导数的 Sobel 近似值检测边缘，在那些梯度最大点返回边缘；

'roberts'　　用导数的 Roberts 近似值检测边缘，在那些梯度最大点返回边缘；

'prewitt'　　用导数的 Prewitt 近似值检测边缘，在那些梯度最大点返回边缘；

'log'　　使用高斯滤波器的拉普拉斯运算对 I 进行滤波，通过寻找 0 相交检测边缘。

[例 15]　对图像“rice.tif”分别采用不同的算子进行边缘检测，结果如图 3-53 所示。

```
%s326.m
I=imread('rice.tif');
BW1=edge(I,'sobel');
BW2=edge(I,'roberts');
BW3=edge(I,'prewitt');
BW4=edge(I,'log');
BW5=edge(I,'canny');
```

```
subplot(231);imshow(I);title('原始图像');
subplot(232);imshow(BW1);title('Sobel 检测边缘图像');
subplot(233);imshow(BW2);title('Roberts 检测边缘图像');
subplot(234);imshow(BW3);title('Prewitt 检测边缘图像');
subplot(235);imshow(BW4);title('LOG 检测边缘图像');
subplot(236);imshow(BW5);title('Canny 检测边缘图像');
```

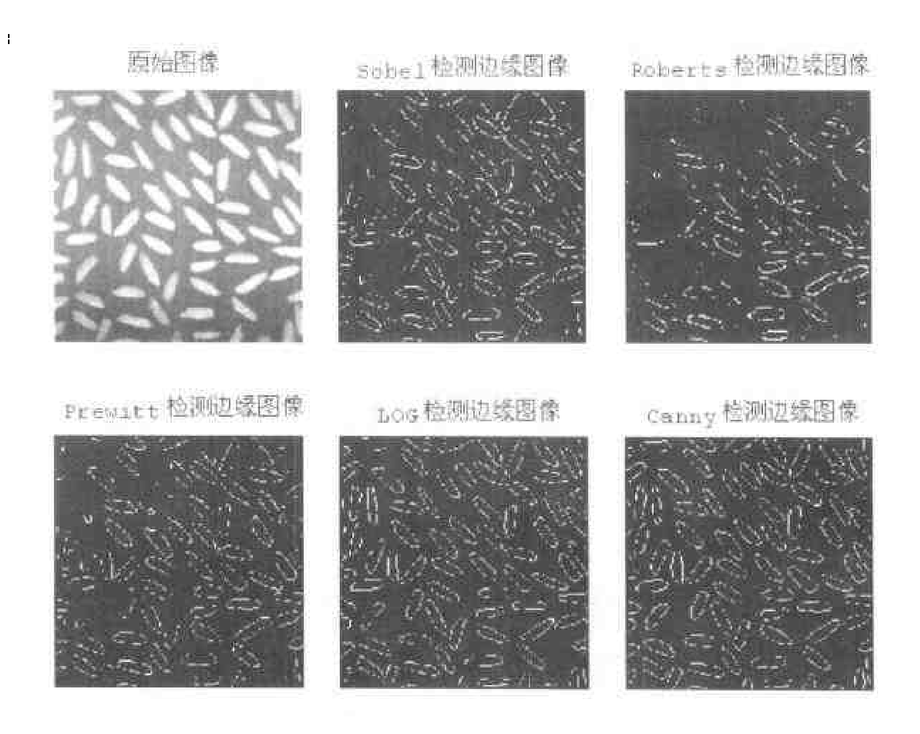


图 3-53　不同算子对图像进行边缘检测的结果

3.2.5　图像的压缩

对于数字图像 $x(m,n)$，$0 \leqslant m < M, 0 \leqslant n < N$，其二维离散余弦变换（DCT）为

$$Y(k,l) = \alpha_k \alpha_l \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \cos\frac{(2m+1)k\pi}{2M} \cos\frac{(2n+1)l\pi}{2N}$$

$$k = 0,1,\cdots,M-1; l = 0,1,\cdots,N-1$$

其中

$$\alpha_k = \begin{cases} 1/\sqrt{M}, k = 0 \\ \sqrt{2/M}, 1 \leqslant k \leqslant M-1 \end{cases}$$

$$\alpha_l = \begin{cases} 1/\sqrt{N}, l = 0 \\ \sqrt{2/M}, 1 \leqslant l \leqslant N-1 \end{cases}$$

二维离散余弦反变换（IDCT）为

$$x(m,n) = \alpha_k \alpha_l \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} Y(k,l) \cos\frac{(2m+1)k\pi}{2M} \cos\frac{(2n+1)l\pi}{2N}$$

$$m = 0,1,\cdots,M-1;\ n = 0,1,\cdots,N-1$$

DCT 变换的实现方法有两种：一种基于 FFT 的快速算法，由函数 dct2 实现；另一种是

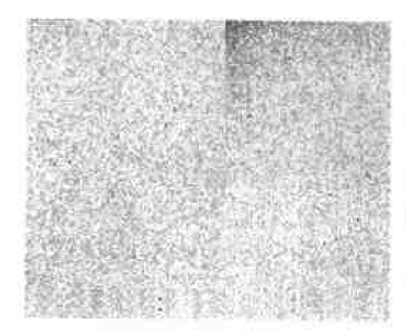
DCT 变换矩阵的方法，由函数 dctmtr 实现。

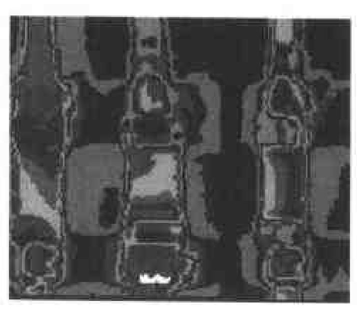
在 JPEG 图像压缩算法中，输入图像被分解成 8×8 或多或 16×16 的块，对每块计算二维 DCT。DCT 系数被量化、编码并传输，而读 JPEG 文件时，解码出 DCT 系数，计算每块的二维反 DCT 变换，然后将每块放到一个图上，达到数字图像压缩的目的。离散余弦变换是 JPEG、MPEG 等数据压缩标准的重要数学基础，在图像处理中有着广泛的应用。

[例 16] 对图像 4pfl.tif 做 DCT 变换及 IDCT 变换。

```
%s339.m
R=imread('4pfl.tif');
I=rgb2gray(R); %RGB 图像变换为灰度图像
J=dct2(I); %DCT 变换
imshow(log(abs(J)),[]);colormap(jet(256));
J(abs(J)<20)=0; %去掉小于一定阈值的变换值，以达到数据压缩目的
K=idct2(J); %反变换重建图像
figure;
subplot(121);imshow(I,[0,255]);
subplot(122);imshow(K,[0,255]);
```

结果如图 3-54 所示。


DCT 变换图像


原图像

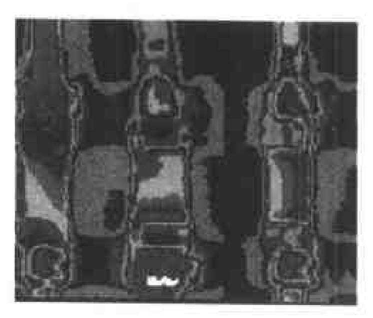
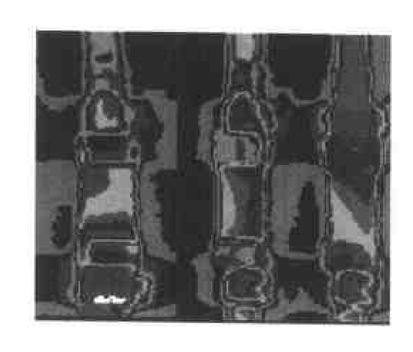

IDCT 反变换图像

图 3-54 原图像与 DCT 和 IDCT 反变换图像比较

[例 17] 对图像 4pfl.tif 进行压缩和解压缩。结果见图 3-55。



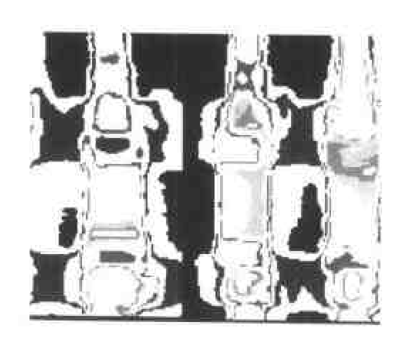


图 3-55 基于 DCT 变换的图像压缩

```
%s340.m
I1=imread('4pfl.tif');
I2=rgb2gray(I1);
```

```
I=double(I2)/255;
T=dctmtx(8);
B=blkproc(I,[8 8],'1*x*2',T,T');
mask=[1 1 1 1 0 0 0 0
      1 1 1 0 0 0 0 0
      1 1 0 0 0 0 0 0
      1 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0];
B2=blkproc(B,[8 8],'1.*x',mask);
I2=blkproc(B2,[8 8],'1*x*2',T',T);
imshow(I),figure;imshow(I2)
```

第 4 章　MATLAB 语言及其使用

4.1　MATLAB 的使用

4.1.1　MATLAB 命令窗口的进入与使用

4.1.1.1　MATLAB 命令窗口的进入

在 Windows 98 下启动 MATLAB 系统后，Windows 98 工作平台上会弹出一个窗口（如图 4-1），这个窗口称为 MATLAB 的命令窗口（MATLAB Command Window）。MATLAB 的命令窗口是用户与 MATLAB 解释器进行通信的工作环境，提示符“|”表示 MATLAB 解释器正等待用户输入命令。所有的 MATLAB 命令、MATLAB 函数，以及 MATLAB 程序都要在这个窗口下运行。

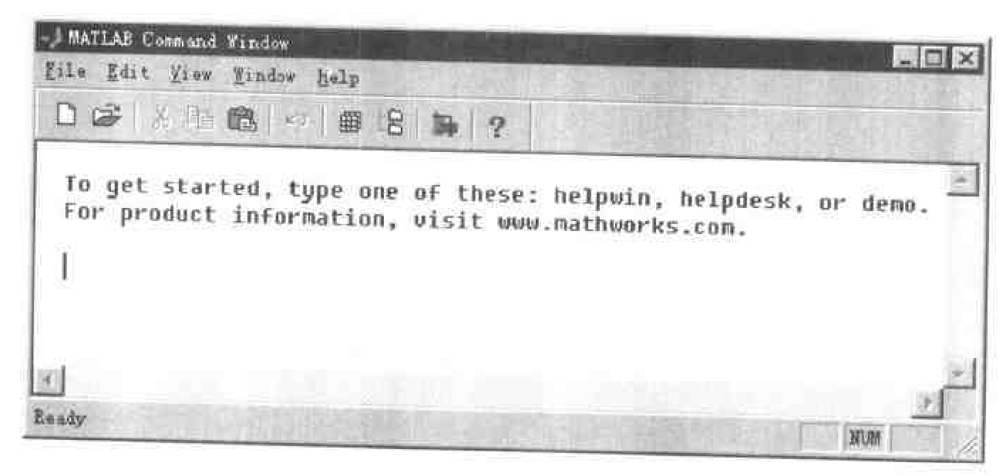


图 4-1　MATLAB 命令窗口

4.1.1.2　MATLAB 命令窗口的使用

例如方程组 A x =b，它的解可直接表示为 x=A\b.

在 MATLAB 的命令窗口输入

```
A=[1,3,2,4.5;2,3.2,4,7;3,4.5,8,9;1,1,2,1]
```

回车后，MATLAB 自动将 A 用矩阵的形式显示出来

```
A= 1.0000   3.0000   2.0000   4.5000
   2.0000   3.2000   4.0000   7.0000
   3.0000   4.5000   8.0000   9.0000
   1.0000   1.0000   2.0000   1.0000
```

然后继续输入

```
b=[1 2 3 4]'
```

回车后，MATLAB 自动将 b 用矩阵（'表示矩阵 b 转置）的形式显示出来

b =

1

2

3

4

最后输入

x=A\b

回车后 MATLAB 给出方程 A x =b 的解，这是一个向量。

x =

6.2759

1.0345

–0.9310

–1.4483

由此例可以看出 MATLAB 给科学计算带来巨大的方便。

4.1.1.3 MATLAB 命令窗口的主要功能

在命令窗口中，除了“File”和“Help”菜单外，其他菜单与 Windows 的一般应用程序相同，下面重点介绍“File”菜单的命令。

- “New”命令有三个选项。“M-file”表示新建一个 M 文件，该命令将打开 MATLAB 的 M 文件编辑/调试器（如图 4-2）；“Figure”新建一个图形窗口；“Model”新建一个模型。

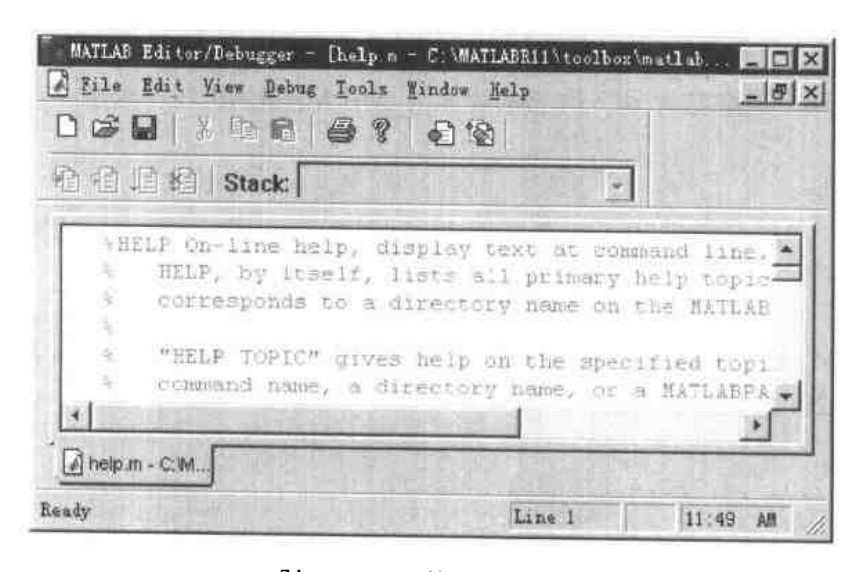


图 4-2 M 文件编辑/调试器

- “Open”命令可以搜寻要打开的 MATLAB 文件所在的目录，然后将其打开。MATLAB 可以打开 M 文件（M 语言程序）、fig 文件（图形文件）、mel 文件（模板文件）。
- “Open Selection”命令在 MATLAB 命令窗口直接选中要编辑的 M 文件，并将其打开。例如，想查看 Help 命令源文件，可以在命令窗口键入“help”，然后选中该命令，原来在“File”菜单中灰色的“Open Selection”将变成“Open help”，单击后 MATLAB 将用 M

文件编辑/调试器将 help.m 文件打开，如图 4-2 所示。

- "Run Script" 命令用于运行脚本文件。
- "Load Workspace" 和 "Save Workspace" 命令用于加载和保存 MATLAB 工作空间的内容。
- "Show Workspace" 命令打开 MATLAB 工作空间浏览器，显示在 MATLAB 中通过命令行所使用变量的集合。每次启动 MATLAB 时，工作空间内没有变量。
- "Show Graphics Property Editor" 命令显示图形属性编辑器。
- "Show GUI Layout Tool" 命令显示图形用户设计向导。
- "Set Path" 命令将打开 MATLAB 路径浏览器。在 MATLAB 中，所有的 M 文件被组织在用户文件系统的目录中，而搜索路径显示了 MATLAB 已经打开的工作目录，只有打开的工作目录下的 M 文件能够在 MATLAB 命令窗口中运行。

4.1.2　MATLAB 的文件编辑/调试器的进入与使用

4.1.2.1　MATLAB 的文件编辑/调试器的进入

MATLAB 的程序可以用编辑器来编写，进入编辑器的方法如下。

① 在 MATLAB 命令窗口直接输入命令

edit test　就会打开编辑器编辑 test．m 文件。

② 在 MATLAB 命令窗口的菜单中选取

- 选择 File 菜单中的 New 或直接按下工具栏上按钮。
- 选择 File 菜单中的 Open 或直接按下工具栏上按钮。

4.1.2.2　MATLAB 的文件编辑/调试器的使用

方程 A x =b，当给定 A 和 b 时它的解 x 可以用 MATLAB 程序求解。在编辑/调试器窗口输入

```
%s201.m
A=[1,3,2,4.5;2,3.2,4,7;3,4.5,8,9;1,1,2,1];
b=[1 2 3 4]';
x=A\b
```

并将其保存为 M 文件 s201.m ，然后在命令窗口输入 s201，则方程的解 x 就会在命令窗口显示出来。若 s201.m 的内容为

```
%s201.m
A=[1,3,2,4.5;2,3.2,4,7;3,4.5,8,9;1,1,2,1];
b=[1 2 3 4];
x=A\b
```

在命令窗口输入 s201 后，命令窗口显示

```
??? Error using ==> \
Matrix dimensions must agree.
Error in ==> C:\数字信号处理\s201.m
On line 4   ==>
```

表明程序 s201.m 有错误，错误在第 4 行，为矩阵维数不匹配。修改后保存再运行直至无误，

即完成编辑、调试的任务。

4.2 MATLAB 数值计算

4.2.1 矩阵运算

矩阵运算是 MATLAB 最基本、最核心的内容。

4.2.1.1 矩阵定义

在 MATLAB 中输入矩阵有三条基本原则：(1) 矩阵元素必须在方括号中；(2) 每行内的元素间用逗号或空格分开；(3) 行与行之间用分号或回车分隔。

如定义矩阵 x=[9,8,7;6,5,4;3,2,1]，这是一个 3 行 3 列的矩阵。

向量是一种特殊的矩阵，它是只有 1 行或只有 1 列的矩阵。在 MATLAB 中定义一个行向量 t ，在命令窗口输入：

```
t=0:0.2:2
```

回车后得到输出结果：

```
t =
  Columns 1 through 7
         0    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000
  Columns 8 through 11
    1.4000    1.6000    1.8000    2.0000
```

表示从 0 开始，到 2 为止，每隔 0.2 取一个数，顺序排成一行构成一个行向量。

4.2.1.2 矩阵的加减运算

矩阵的加减运算表示为：A+B、A-B。要求 A、B 具有相同的维数，或其中一个是标量。

如输入：

```
A=[1,2,3;4,5,6;7,8,9];
B=[2,3,4;5,6,7;8,9,0];
C=A+B
```

得到 C =

```
     3     5     7
     9    11    13
    15    17     9
```

输入：D=A-B

得到 D =

```
    -1    -1    -1
    -1    -1    -1
    -1    -1     9
```

从此例可见，矩阵的加减运算是两个矩阵每一个元素对应相加减。当标量与矩阵加减运算时，该标量与矩阵每一个元素相加减。

4.2.1.3 矩阵的乘除及乘方运算

矩阵的乘除及乘方运算表示为：A*B、A/B、A\B 、A^p。矩阵相乘时要求 A 的列数等

于 B 的行数，即满足矩阵相乘的条件。矩阵相除是用来表达线性方程的解，A/B 时要求 A、B 列数相同；A\B 时要求 A、B 行数相同。矩阵的乘方要求矩阵是方阵，且 p 只能是标量。

输入：A=[1,2,3;4,5,6];

B=[1,2,3]';

C=A*B

得到 C =

14

32

输入：A=[1,2,3;4,5,6];

B=[1,2,3];

D=A/B

得到 D =

1.0000

2.2857

4.2.1.4 矩阵元素的提取

在 MATLAB 中 A(:,2)表示矩阵第 2 列的元素，冒号表示所有的行。A(3,:)表示矩阵第 3 行的所有元素。

输入：A=[1,2,3;4,5,6;7,8,9];

A(3,:)

得到 ans =7 8 9

4.2.2 数组运算

数组是 MATLAB 中一种重要的数据类型，它的定义方法和矩阵一样，但在计算上与矩阵有很大区别，但本质上是一致的。

4.2.2.1 数组的定义

定义数组 B=[1,2,3;4,5,6] （定义二维数组）

B =

1 2 3

4 5 6

B(1,3)=100% （修改数组元素）

B =

1 2 100

4 5 6

一行一列的数组就是一个数。

4.2.2.2 数组的计算

数组的加减运算表示为：A+B、A–B。要求 A、B 具有相同的维数，或其中一个是标量。数组的乘除及乘方运算表示为：A.*B、A./B、A.\B 、A.^p。

```
A=[1,2,3;4,5,6];
B=[3,4,5;7,8,9];
A+B
ans =
     4     6     8
    11    13    15
```

可以看到数组的加减与矩阵的加减没有区别，所以表示相同。

```
A*B
??? Error using ==> *
Inner matrix dimensions must agree.
```

系统告诉我们这两个矩阵的维数不匹配，这是由于把 A、B 看成矩阵，它们是不能相乘的。

将 A、B 看成是数组，却可以进行相乘，结果仍然是一个 2×3 的数组，其中每一个元素是 A、B 中对应元素的乘积。

```
A.*B
ans =
     3     8    15
    28    40    54
```

MATLAB 为了区分数组和矩阵运算，专门定义：

数组乘法“.*”—— A、B 中对应元素的乘积；

数组除法“./”、“ .\”—— A、B 中对应元素的相除；

数组乘方“.^”——每个元素都乘方。

4.2.2.3 数组的关系和逻辑运算

（1）数组的关系运算

<　　小于

<=　 小于等于

==　 等于

>　　大于

>=　 大于等于

~=　 不等于

在关系运算中，当关系成立时结果为 1（真），不成立时结果为 0（假）。

（2）数组逻辑运算

&　与

|　或

~　非

在逻辑运算中，只有非零（1）和 0 两种元素。

4.2.3 多项式运算

多项式在工程计算中有着广泛的应用，很多特殊的曲线都用多项式来拟合它的方程。

4.2.3.1 多项式的定义

（1）直接定义

多项式 $P(n)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$

在 MATLAB 中用行向量表示：P=[a_0,a ,...,a_{n-1},a_n]

（2）用命令 poly 创建

如果 A 是矩阵，则 poly(A)将创建 A 矩阵的特征多项式；如果 A 是向量，则 poly(A)将创建以 A 中各元素为根的多项式。

定义多项式 a（向量定义）、b（poly 函数定义）：

```
a=[1,2,3]
a =  1      2      3
b=poly([1,2,3])
b = 1     -6     11     -6
```

得到以下两个多项式：

$$a(x)=x^2+2x+3$$
$$b(x)=x^3-6x^2+11x-6$$

4.2.3.2 多项式的运算

多项式运算函数见表 4-1。

表 4-1 多项式运算函数

函　数	功　能	函　数	功　能
Roots	多项式求根	Polyfit	多项式曲线拟合
Poly	由根创建多项式	Polyder	多项式求导
Polyval	多项式求值	Conv	多项式相乘（卷积）
Residue	部分分式展开（求留数）	Deconv	多项式相除（解卷）

● 多项式求根

```
roots(a)  %求多项式 a 的根
ans =  -1.0000 + 1.4142i
       -1.0000 - 1.4142i
```

● 多项式求值

```
polyval(b,6)  %求多项式 b 在 x=6 这点的值
ans  =  60
```

● 多项式拟合和求值

例如，某运算放大器输出电压随频率变化的实测值见表 4-2，利用多项式拟合和求值函数，计算此运算放大器的通频带。

表 4-2 运算放大器输出电压随频率变化的测量数据

频 率/Hz	1	5	10	20	50	80	90	100	120	130	150	160	180	190
输出电压/V	6.0	6.0	6.0	6.0	5.9	5.4	5.3	5.1	4.8	4.8	4.3	4.3	4.1	3.9

```
%s202.m
f=[1,5,10,20,50,80,90,100,120,130,150,160,180,190];
u0=[6.0,6.0,6.0,6.0,5.9,5.4,5.3,5.1,4.8,4.8,4.3,4.3,4.1,3.9];
```

```
u1=polyfit(f,u0,1) %一阶方程曲线拟合
u3=polyfit(f,u0,3) %三阶方程曲线拟合
f1=1:0.1:200; %频率向量
u10=polyval(u1,f1); %计算一阶拟合多项式的值
u30=polyval(u3,f1) %计算三阶拟合多项式的值
plot(f,u0,'*',f1,u10,f1,u30);grid %画出曲线
```

执行后得到

u1 = – 0.0116　　6.1979

u3 = 0.0000　 – 0.0002　　0.0037　　5.9937

即得多项式 $u1(f)=-0.0116f+6.1979$

$u3(f)=-0.0002f^2+0.0037f+5.9937$

由 $u30$ 可知，当 f=164Hz 时，$u30$=4.2（V），故此运算放大电器的通频带为 164Hz。从图 4-3 可以看出三阶拟合的效果比一阶拟合的效果好。

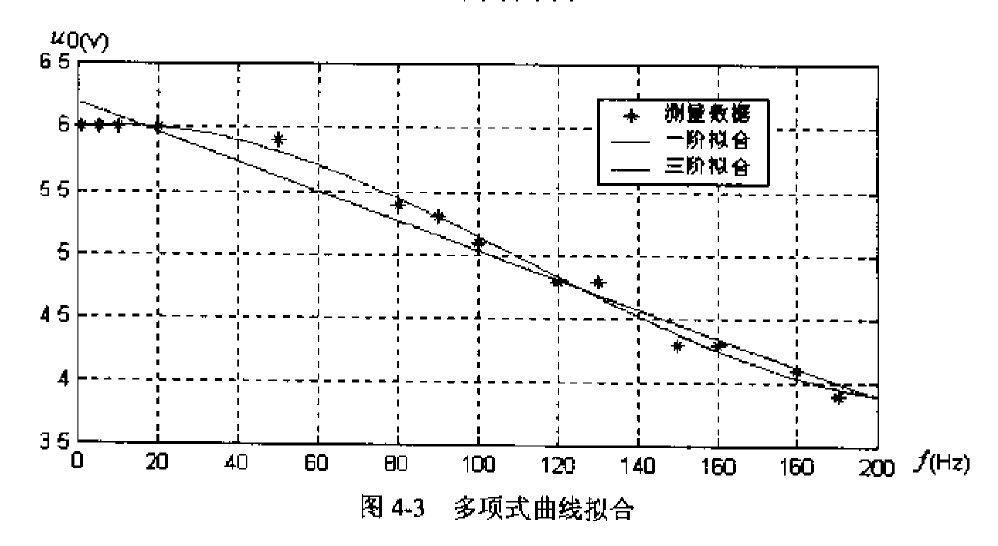


图 4-3　多项式曲线拟合

- 多项式相乘

多项式相乘对应于卷积运算　$c(s)=\sum_{j=1}^{s}a(j)b(s+1-j)$

例如多项式 $a(s)=s^2+2s+3$ 和 $b(s)=4s^2+5s+6$，计算它们的乘法。

输入 a=[1,2,3];b=[4,5,6];c=conv(a,b)

得到 c =4　　13　　28　　27　　18

可以写出：$c(s)=4s^4+13s^3+28s^2+27s+18$

4.3　MATLAB 符号计算

在工程和科学上经常遇到符号计算的问题，如微分、积分、线性代数运算、求解方程和方程组及积分变换等。

4.3.1　符号运算的基本操作

4.3.1.1　创建符号和表达式

sym 命令创建符号和表达式，同时创建多个符号时用 syms 命令。如

输入 x=sym('alpha')

rho=sym('(1+sqrt(5))/2')

得到 x=

alpha

rho =

(1+sqrt(5))/2

表明符号 x 代表字符串 alpha，符号 rho 代表表达式 $\frac{1+\sqrt{5}}{2}$。

4.3.1.2 创建符号矩阵

用[ ]创建符号矩阵。如

```
syms a b c%一次性创建变量符号 a、b、c
a='alpha'; %给 a 赋值
b='beta'; %给 b 赋值
A=[a b c;b c a;c a b] %创建符号矩阵
A =
   [ alpha,  beta,     c]
   [  beta,     c, alpha]
   [     c, alpha,  beta]
```

4.3.1.3 创建实型和复数型符号变量

sym 命令可以确定符号的变量属性，通过 real 选项实现。如

```
syms a b real
c=a+i*b
```

这里 a、b 为实型符号变量，c 为复数型符号变量。

4.3.1.4 符号和数值的转换

sym 命令有四种选项用于将数值转换为符号表达式：

```
rho=(1+sqrt(5))/2;
```

● 选项 f：返回符号的浮点表示。

```
sym(rho,'f') %将 rho 的值改为浮点数输出
ans ='1.9e3779b97f4a8'*2^(0)
```

● 选项 r：返回符号的有理数形式。

```
sym(rho,'r')
ans =7286977268806824*2^(-52)
```

● 选项 e：返回符号的有理数形式，同时根据 eps（浮点运算的相对精度）给出理论表达式和实际计算值的误差。

```
sym(0.333,'e')
ans =333/1000+41*eps/500
```

● 选项 d：返回十进制小数，有效位数由 digits 定义。

```
digits(10) %设置有效位数为 10
sym(rho,'d')
```

```
ans =1.618033989
```

4.3.2 微分与积分运算

4.3.2.1 符号变量的确定

MATLAB 中数学函数的表示见表 4-3。

表 4-3 MATLAB 中数学函数的表示

数学函数	$F(x)=x^n$	$G(x)=\sin(at+b)$	$H(x)=e^{2x}$
MATLAB 中的表示	F=x^n	G=sin(a*t+b)	H=exp(2*x)

通常独立变量是小写字母且靠近字母表的后面（如 x、y、z），函数 **findsym** 将给出符号表达式中隐含的符号变量。

Findsym 的规则：符号表达式中的缺省符号变量为靠近小写字母 x 的优先，在 x 后面的优先。

4.3.2.2 微分运算

diff 函数可以指定符号表达式和求导的符号变量以及求导的阶数。

● 如 $g=\dfrac{xy}{x^2+y^2}+\lg x\sin y$，求 $\dfrac{d^2g}{dx^2}$。

```
syms x y %定义符号变量
g=x*y/(x^2+y^2)+log(x)*sin(y); %定义符号表达式
diff(g,x,2) %对函数 g 求 x 的二阶导数
ans =-6*y/(x^2+y^2)^2*x+8*x^3*y/(x^2+y^2)^3-1/x^2*sin(y)
```

4.3.2.3 积分运算

int 函数可以对符号表达式进行积分运算。

● 求表达式的不定积分，计算 $\int\dfrac{1}{x^2+dx}$

```
int(1/(x^2+1))
ans =atan(x)
```

● 求函数的定积分，计算 $\int_0^{\frac{2\pi}{3}}\cos x dx$

```
int(cos(x),x,0,pi*2/3)
ans = 1/2*3^(1/2)
```

4.3.2.4 求和运算

当符号变量的和存在时，用 symsum 求和。

● 求-1+1/2-1/3+1/4-1/5+…

```
syms k
symsum((-1)^k/k,1,inf)
ans =-log(2)
```

即结果为–log(2)。

4.3.3 求解方程和方程组

用 solve 求解代数方程和方程组，dsolve 求解微分方程和方程组。

4.3.3.1 简单代数方程

● 求解 $ax^2+bx+c=0$。

```
syms a b c x s %赋值求解
s=a*x^2+b*x+c;
solve(s)
ans =[ 1/2/a*(-b+(b^2-4*a*c)^(1/2))]
     [ 1/2/a*(-b-(b^2-4*a*c)^(1/2))]
```

在这里将表达式赋给变量 s，solve(s)缺省将表达式的值认为是 0。

● 求解 $2x^2+3x+4=e^x$。

```
solve('2*x^2+3*x+4=exp(x)') %直接求解，必须用单引号
ans =3.7839093023
```

4.3.3.2 代数方程组

解一个二元二次方程组

$$xy=3$$
$$x^2+y=4$$

解：

```
[x,y]=solve('x*y=3','x^2+y=4') %解代数方程组
 x =[  1][ -1/2+1/2*13^(1/2)][ -1/2-1/2*13^(1/2)]
 y =[  3][ 1/2+1/2*13^(1/2)][ 1/2-1/2*13^(1/2)]
```

4.3.3.3 微分方程和微分方程组

使用 dsolve 求解微分方程，其格式与 solve 类似，符号表达式中的微分符号用 D 表示，D2、D3、…DN 分别对应二阶微分、三阶微分、…N 阶微分。

● 求解常微分方程

$$\frac{d^2x}{dt^2}+2\frac{dx}{dt}+x=\cos(t)$$

```
dsolve('D2x+2*Dx+x=cos(t)')
ans = 1/2*sin(t)+C1*exp(-t)+C2*exp(-t)*t
```

其中 C1 和 C2 表示常数。

● 求解常微分方程组

$$\frac{dx}{dt}=3x+4y$$
$$\frac{dy}{dt}=-4x+3y$$

不包含初始值输入

```
[x,y]=dsolve('Dx=3*x+4*y','Dy=-4*x+3*y')
```

得到解 x =exp(3*t)*(cos(4*t)*C1+sin(4*t)*C2)

y =exp(3*t)*(-sin(4*t)*C1+cos(4*t)*C2)

包含初始值输入

[x,y]=dsolve('Dx=3*x+4*y','Dy=-4*x+3*y','x(0)=0','y(0)=1')

得到解 x =exp(3*t)*sin(4*t)

y =exp(3*t)*cos(4*t)

4.3.4 积分变换

积分变换和其离散形式在工程和科学研究领域起着重要作用，尤其是傅里叶（Fourier）变换在数字信号处理中得到广泛应用。

4.3.4.1 Fourier 变换和 Fourier 反变换

Fourier 变换用 fourier 函数表示，Fourier 反变换用 ifourier 表示。

● Fourier 变换

```
%s203.m
syms x u v
f=x^2+exp(u);
F1=fourier(f)
F2=fourier(f,v)
F3=fourier(f,u,v)
F1 =-2*pi*Dirac(2,w)+2*exp(u)*pi*Dirac(w)
F2 =-2*pi*Dirac(2,v)+2*exp(u)*pi*Dirac(v)
F3 =2*x^2*pi*Dirac(v)+fourier(exp(u),u,v)
```

说明：（1）$F1$ 是求函数对缺省自变量 x 的 Fourier 变换，返回 w 的函数。

（2）$F2$ 是求函数对指定自变量 v 的 Fourier 变换，返回 v 的函数。

（3）$F3$ 是求函数对指定自变量 u 的 Fourier 变换，返回 v 的函数。

● Fourier 反变换

4.3.4.2 Laplace 变换和 Laplace 反变换

Laplace 变换用 laplace 函数表示，Laplace 反变换用 ilaplace 表示。

4.3.4.3 z 变换和 z 反变换

z 变换用 ztrans 函数表示，z 反变换用 iztrans 表示。

4.4 MATLAB 绘图功能实例

在科学计算中往往需要处理机大量的数据，如果把这些数据用图形表现出来就能很容易地发现这些数据的内在联系，大大提高工作的效率。MATLAB 的绘图功能包括二输绘图和三维绘图。

4.4.1 MATLAB 的图形窗口

在 MATLAB 下建立一个图形窗口由命令 figure 完成，每执行一次 figure 命令就产生一个图形窗口，可以同时产生若干个图形窗口，MATLAB 自动在这些窗口的名字上添加

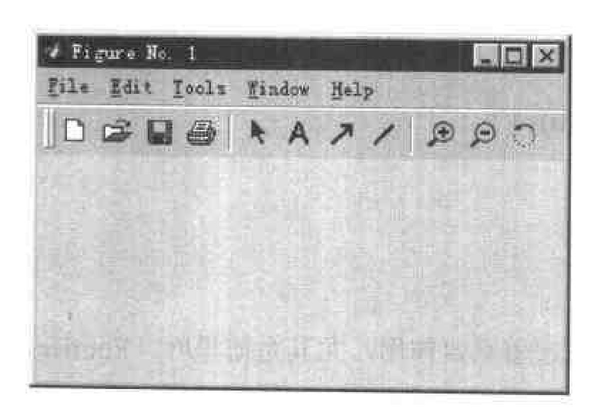


图 4-4 figure 命令产生的图形窗口

序号作为区别。

图 4-4 是用 figure 命令产生的图形窗口。在产生图形基础上增加了菜单和工具条。图形窗口工具条上有几个专用按钮，是针对图形进行操作的，其功能分别为：

按钮：允许对图形进行编辑；

按钮：在图形窗口中添加文本；

按钮：在图形窗口中添加箭头；

按钮：在图形窗口中添加直线；

按钮：允许对图形进行缩放操作；

按钮：允许把图形旋转为三维图形。

关闭窗口由 close 命令完成，每执行一次 close 命令关闭一个当前的图形窗口，要同时关闭所有窗口，用 close all 来完成。

4.4.2 二维图形绘制

4.4.2.1 基本二维图形绘制函数 plot

plot 是基本的绘图命令，输入 plot（a,b）这个命令就会画出一个以 *a* 为 X 轴、*b* 为 Y 轴的图形。如

```
x=0:pi/20:2*pi;
plot(x,sin(x));
```

绘制如图 4-5 所示的一个周期的正弦曲线。

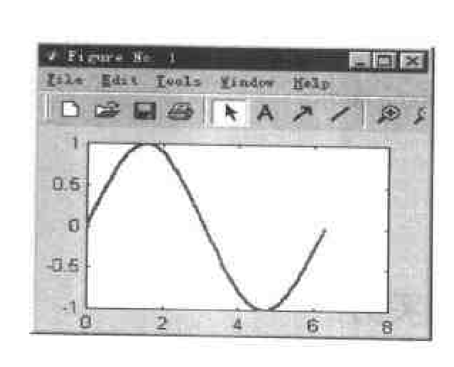


图 4-5 plot 函数绘制的曲线

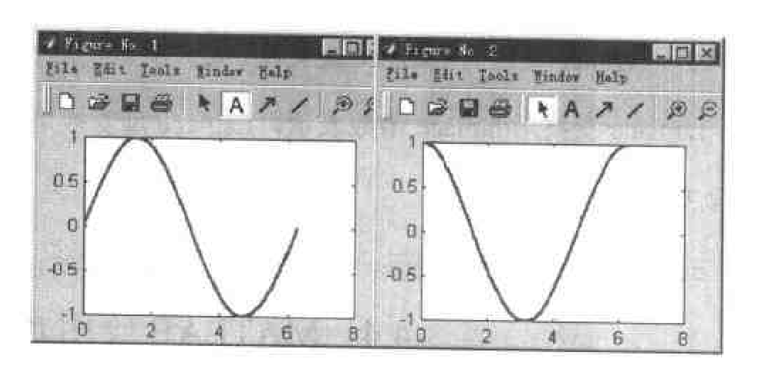


图 4-6 figure 函数绘制的两张图形

4.4.2.2 figure

figure 是成像的意思，有时候画了几张图，而屏幕上所能看到的只是最近打开的那张。如果想回头看第一张就可以输入 figure 命令。如

```
%s204.m
```

```
x=0:pi/20:2*pi;
figure(1);
plot(x,sin(x));
figure(2);
plot(x,cos(x));
```

绘制如图 4-6 所示的一个周期的正弦、余弦曲线。

4.4.2.3 subplot

subplot(m,n,i)是图形显示时分割窗口的命令，把窗口分为 m 行 n 列子窗口，第 i 个窗口为当前窗口。

```
%s205.m
x=0:pi/20:2*pi;
subplot(2,2,1);plot(x,sin(x));
subplot(2,2,2);plot(x,cos(x));
subplot(2,2,3);plot(x,x);
subplot(2,2,4);plot(x,-x);
```

在一个图形中绘制图 4-7 所示的四个部分。

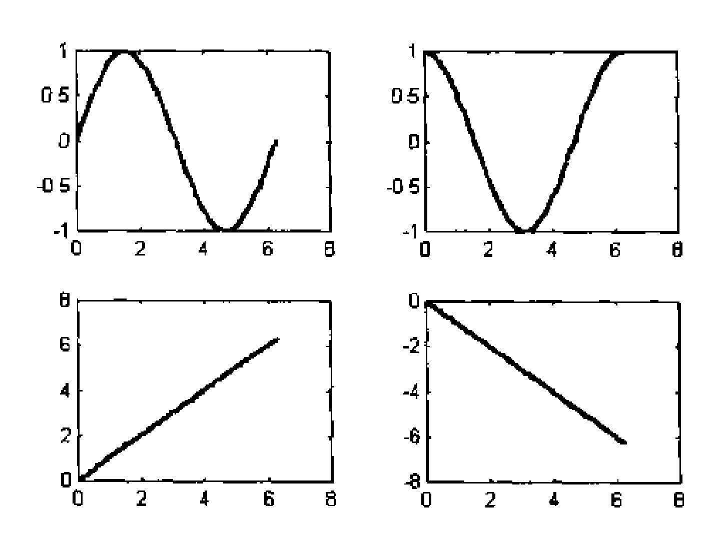


图 4-7 subplot 函数对窗口的分割

4.4.2.4 指定颜色、线条样式和标记符号

plot 命令可多加一个参数用来控制图形的颜色、粗细等。如

```
x=0:pi/20:2*pi;
plot(x,sin(x),'r+:');
```

绘制图 4-8 所示的图形，其参数含义如表 4-4 所示。

可以同时在一个图形上画出几条曲线，输入如下命令。

```
%s206.m
x=0:pi/20:2*pi;
plot(x,sin(x),'r+:',x,cos(x),'b');
```

绘制两条曲线如图 4-9 所示。

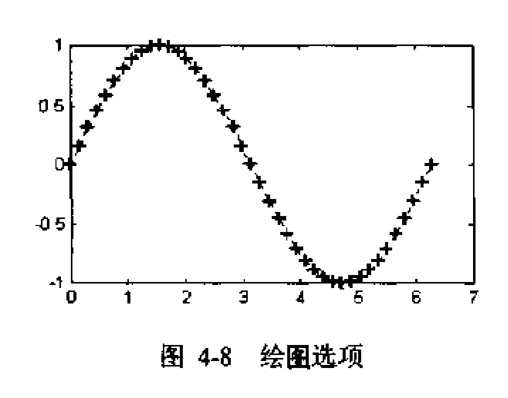


图 4-8　绘图选项

表 4-4　常用绘图参数的含义

参数	含义	参数	含义	参数	含义
y	黄色	o	圆	-.	点划线
m	紫色	x	打叉	——	破折线
c	青色	+	加号	^	向上的三角形
r	红色	*	星号	<	向左的三角形
g	绿色	s	正方形	>	向右的三角形
b	蓝色	d	菱形	p	五角星形
w	白色	v	向下的三角形	h	六角星形
k	黑色	-	实线		
.	点	:	虚线		

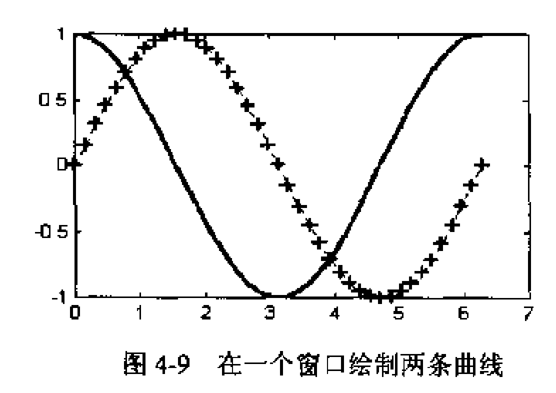


图 4-9　在一个窗口绘制两条曲线

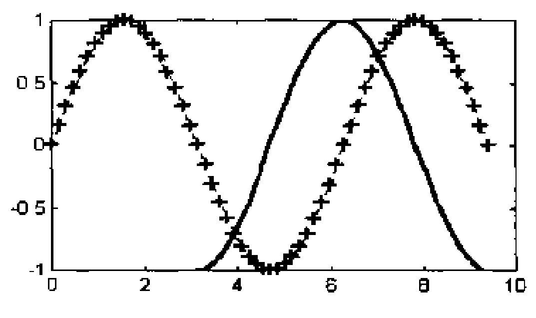


图 4-10　两条曲线具有不同的横坐标

在一个图形上画出几条曲线，而且每条曲线的横坐标可以不同，每一组向量也都可以有不同的长度。程序如下。

```
%s207.m
x1=0:pi/20:3*pi;
x2=pi:pi/20:3*pi;
plot(x1,sin(x1),'r+:',x2,cos(x2),'b');
```

绘制图 4-10 两条曲线，它们的坐标位置不同而且长度也不同。

4.4.2.5　标题与轴标签

在图形上加标题用 title 命令。如：

```
%s208.m
x=0:pi/20:2*pi;plot(x,sin(x));
title('sine wave');
```

绘制图 4-11 曲线，在图形上加标题。title 要写在 plot 之后，否则不起作用。

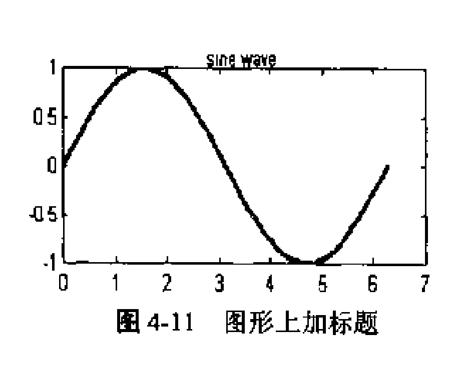


图 4-11　图形上加标题

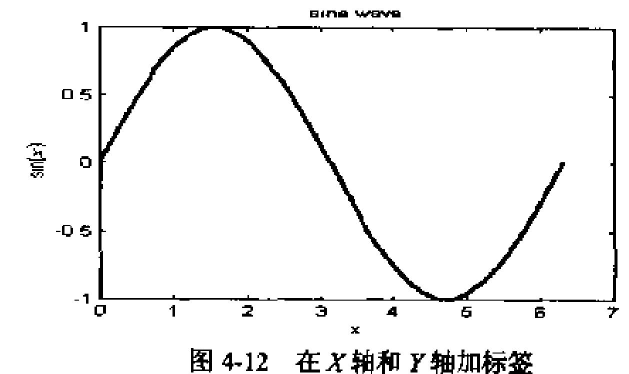


图 4-12　在 X 轴和 Y 轴加标签

在 X 轴和 Y 轴上加标签用 xlabel 与 ylabel 命令。

```
%s209.m
x=0:pi/20:2*pi;
plot(x,sin(x));
title('sine wave');
xlabel('x');
ylabel('sin(x)');
```

绘制曲线如图 4-12，在 X 轴和 Y 轴上分别加上了标签。

4.4.2.6 轴间距的设定

在指数或对数的图形中常用以 log 为坐标轴进行表示，可用命令 loglog、semilogx、semilogy 命令。如：

```
%s210.m
x=0:1:1000;
subplot(1,3,1);loglog(x,x); %x轴对数_y轴对数
subplot(1,3,2);semilogx(x,x); %x轴对数
subplot(1,3,3);semilogy(x,x); %y轴对数
```

绘制对数曲线如图 4-13 所示。

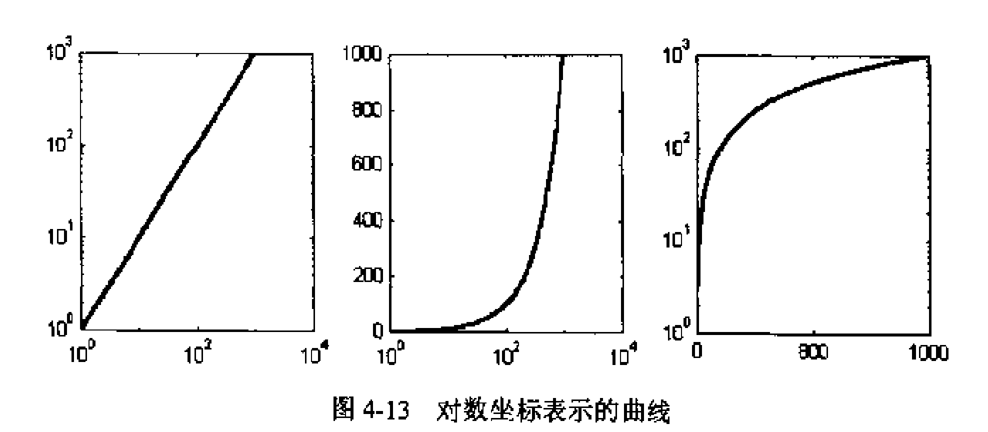


图 4-13　对数坐标表示的曲线

4.4.2.7 用 axis 设定轴的范围

MATLAB 总是把 X，Y 轴的范围设定得比图形的最大值还要大一些，如果要求图形大小刚刚取齐，或希望轴的范围由自己决定，可以利用 axis 命令。axis 的用法为 axis（[X 最小值，X 最大值，Y 最小值，Y 最大值]）。如：

```
%s211.m
x=0:pi/20:2*pi;
plot(x,sin(x));
axis([0,2*pi,-1,1]);
```

绘制曲线如图 4-14 所示。

4.4.2.8 text

MATLAB 可以用 text 命令在图中任意特定位置上作标示，text 命令的用法为：
text（X 值，Y 值，'想加的标示'）。

```
%s212.m
x=0:pi/20:2*pi;
plot(x,sin(x));
axis([0,2*pi,-1,1]);
text(3*pi/4,sin(3*pi/4),'\leftarrowsin(x)=0.707');
```

绘制曲线如图 4-15，所加标示为“←sin(x)=0.707”。

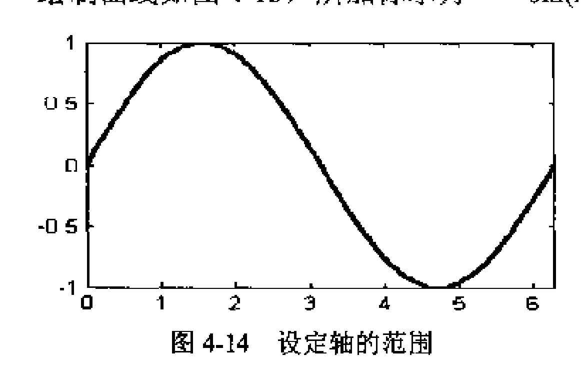


图 4-14 设定轴的范围

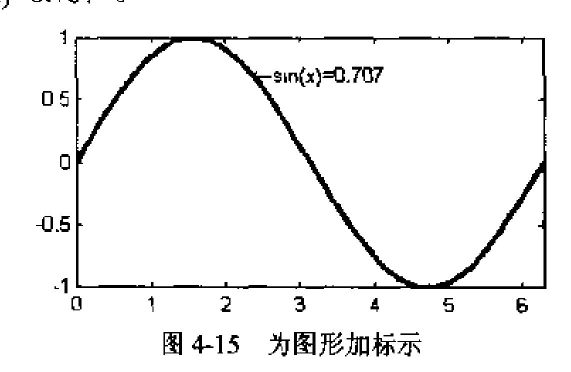


图 4-15 为图形加标示

4.4.2.9 hold on 与 hold off

如果在原来的图上加画其他线条或文字，而不把原来的图覆盖掉，用 hold on 命令。

```
%s213.m
x=0:pi/20:2*pi;
plot(x,sin(x));
axis([0,2*pi,-1.2,1]);
hold on
plot(x,sin(x)-0.2,'r+:');
```

绘制曲线如图 4-16 所示。

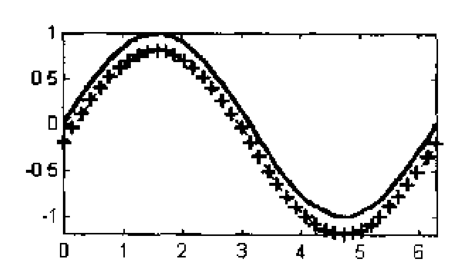


图 4-16 在原来的图上加其他线条或文字

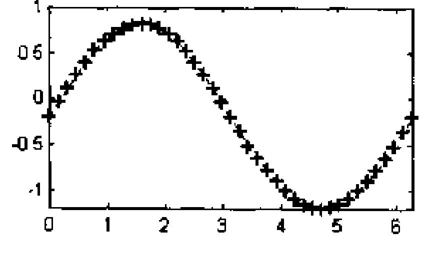


图 4-17 把新图覆盖在旧图上

如果想取消 hold on，把新图直接覆盖到旧图上，用 hold off。

```
%s214.m
x=0:pi/20:2*pi;
plot(x,sin(x));
hold off
plot(x,sin(x)-0.2,'r+:');
axis([0,2*pi,-1.2,1]);
```

绘制曲线如图 4-17 所示。

4.4.2.10 grid on

用虚线来标示各坐标。

```
%s215.m
x=0:pi/20:2*pi;
plot(x,sin(x));
axis([0,2*pi,-1.2,1]);
hold on
plot(x,sin(x)-0.2,'r+:');
grid on
```

绘制曲线如图 4-18 所示。

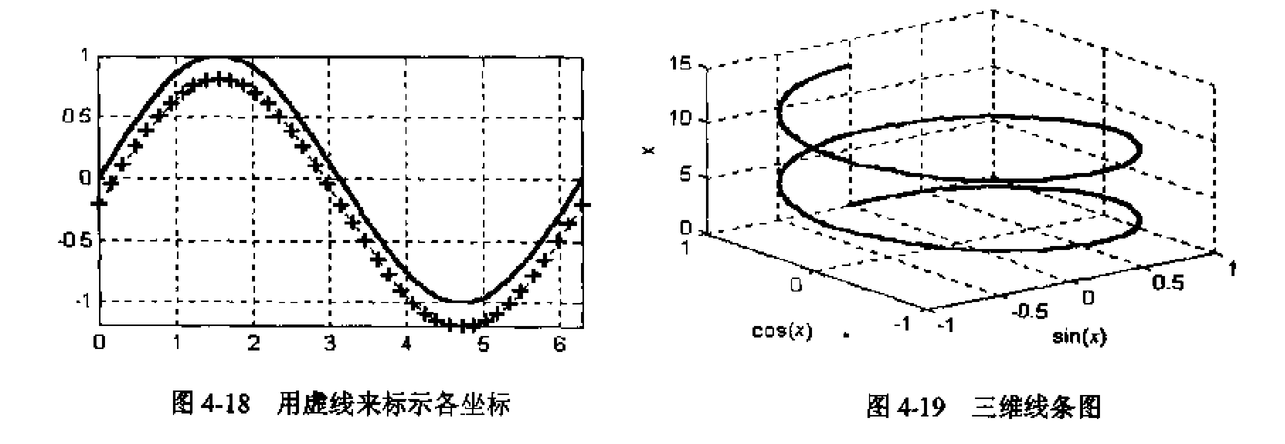


图 4-18 用虚线来标示各坐标　　图 4-19 三维线条图

4.4.3 三维图形

4.4.3.1 线性 3D 图(plot3)

建立三维线条图用 plot3 命令。plot3 的一般调用格式为：

```
plot3(x,y,z,s)
```

式中 x、y、z 分别为第一到三维数据，s 是设置线型、颜色、数据点标记的字符串。如：

```
%s216.m
x=0:pi/20:4*pi;
plot3(sin(x),cos(x),x);
grid on
```

绘制三维线条图如图 4-19 所示。

4.4.3.2 网状 3D 图(mesh)

网格图是指把相邻的数据点连接起来形成网状曲面，建立网格图用 mesh 函数。如

```
%s217.m
z=peaks(50); % peaks (N) 的作用是取一个N×N的高斯分布矩阵
h=mesh(z);% mesh 画网状 3D 图
```

绘制网格图如图 4-20 所示。

```
h=meshc(z); % meshc 画带有基本等高线的网状 3D 图
```

绘制带有基本等高线的网格图如图 4-21 所示。

```
h=meshz(z); % meshz 画带有基准平面的网状 3D 图
```

绘制带有基准平面的网格图如图 4-22 所示。

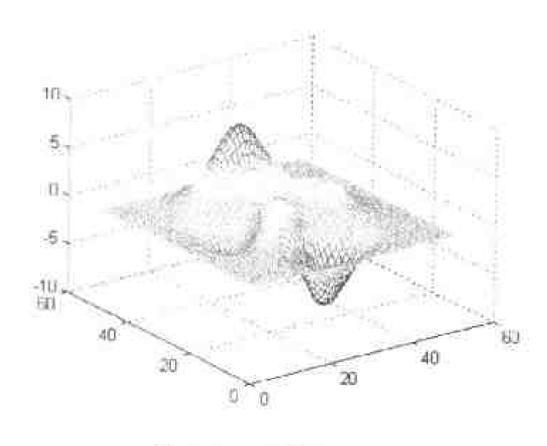


图 4-20　网格图

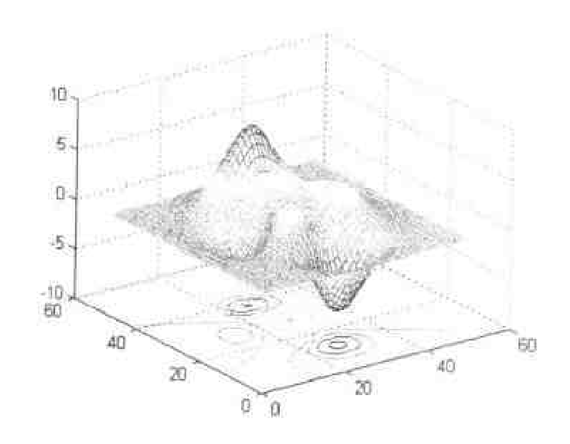


图 4-21　带有基本等高线的网格图

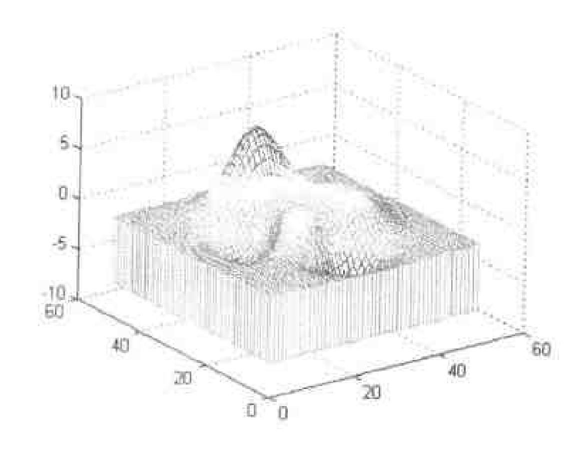


图 4-22　带有基准平面的网格图

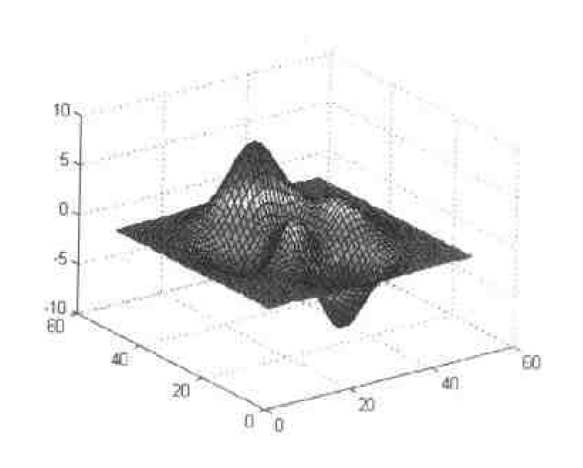


图 4-23　面状图

4.4.3.3　面状 3D 图(surf)

表面图是指把网格图表面的网格围成的小片区域用不同的颜色填充形成的彩色表面。建立表面图用 surf 函数。如：

```
%s218.m
z=peaks(50);
surf(z); % surf 画面状 3D 图
```

绘制面状图如图 4-23 所示。

4.4.3.4　设定颜色（colormap）

可以用颜色映像（colormap）选择喜欢的色彩分布。如：

```
%219.m
z=peaks(50);
subplot(1,2,1);h=surf(z);
subplot(1,2,2);h=mesh(z);
colormap hot;
```

绘制的图形如图 4-24 所示。

其他颜色设定还有：hsv、gray、pink、cool、bone、copper 等。

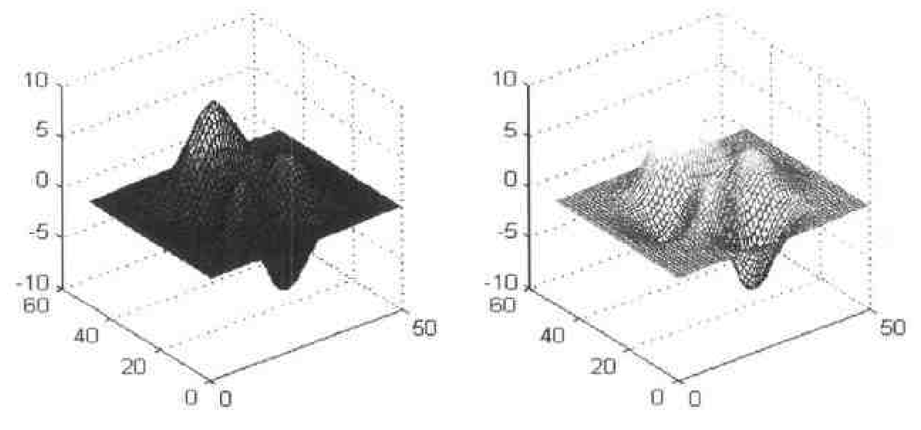


图 4-24 颜色映像（colormap）

4.4.3.5 设定光线来源(light)

light 这个命令可以设定光源的方向。如：

```
%s220.m
z=peaks(50);
h=mesh(z);
light('position',[0 0 5]);
```

绘制的图形如图 4-25 所示。

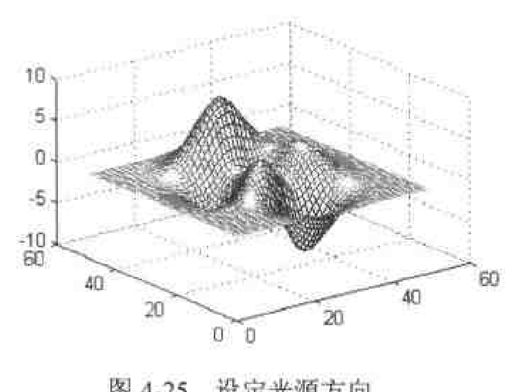


图 4-25 设定光源方向

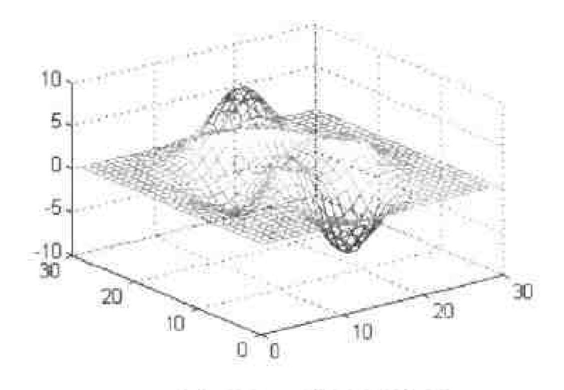


图 4-26 产生透视效果

4.4.3.6 透视(hidden off)

在 mesh 命令后面加 hidden off，可使网状图产生透视效果。

```
%s221.m
z=peaks(30);
h=mesh(z);
hidden off;
```

绘制的图形如图 4-26 所示。

4.4.4 特殊图形

4.4.4.1 等高线图（等温线图）

等高线图最常用于地理勘测中的地形标绘。在 MATLAB 中，用于绘制等高线图的函数

有：clabel，contour，contour3，contourf。函数 meshc 和 surfc 在绘制网格图和表面图时也同时绘制等高线。

函数 contour 和 contour3 分别用来绘制二维和三维等高线图。它们能够自动根据 z 值的最小最大值来确定等高线的条数，也可以根据给定的参数来取值。这两个函数还可以返回每条等高线的句柄和高度值。利用这些返回值，clabel 函数可以给等高线作标注。函数 contourf 与 contour 一样，用来为二维图形作标注，但它在等高线之间填充不同的颜色。

下面举例概括上面所列举的方法。

```
%s222.m
z=peaks;
[c,h]=contourf(z,5);
clabel(c,h);
```

绘制的图形如图 4-27 所示。

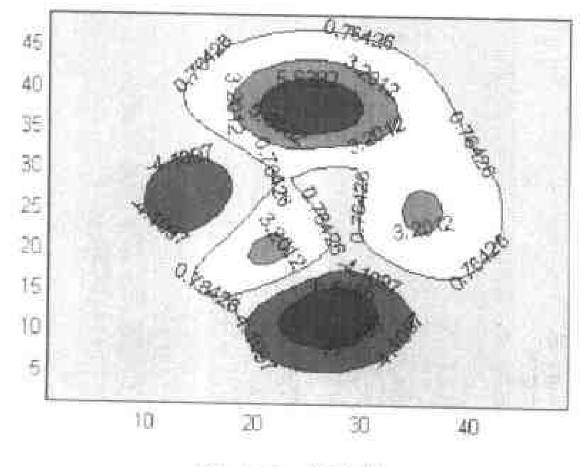


图 4-27 等高线

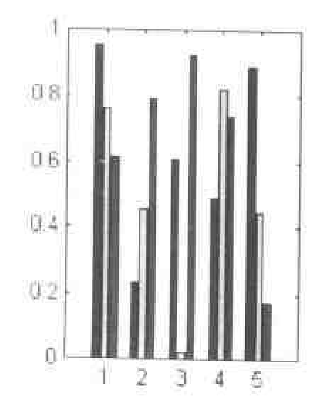
4.4.4.2 直方图和面积图

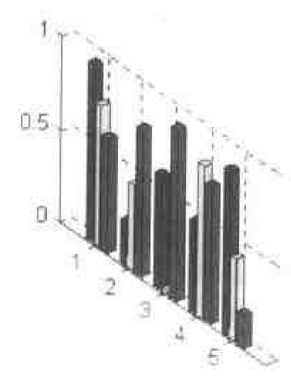
直方图和面积图是平常工作中经常用到的两种图形。它们适用于对不同数据的比较，以及分析各个数据在总体中所占的比例。面积图是用于连续数据，而直方图则是用于离散的数据。

MATLAB 中用于绘制直方图的函数有四种：bar，bar3，barh，bar3h。这当中，bar 和 bar3 用于绘制二维和三维的竖向直方图，而 barh 及 bar3h 用于绘制横向直方图。这些函数都是以输入数据矩阵的每一列为一组数据，并以相同的颜色表示，而把矩阵的行画在一起。举一个例子进行说明。

```
%s223.m
y=rand(5,3); %产生一组随机数矩阵
subplot(1,3,1);bar(y); %画直方图
subplot(1,3,2);bar3(y,'group'); %即缺省形式的直方图
subplot(1,3,3);bar(y,'stack'); %将同一列数据画在一个直方条上
```

绘制的图形如图 4-28 所示。





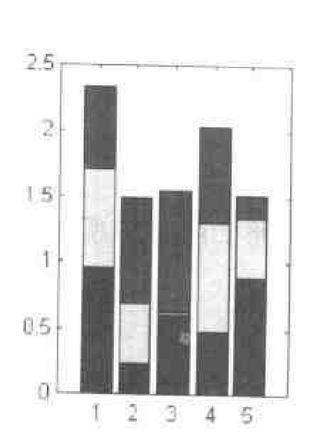


图 4-28 直方图

在上例中，我们使用了两个选项："stack"和"group"。"stack"选项的作用是将同一列数据画在一个直方条上，这样可以直观地观察每一个数据占该列数据总和的比例。"group"选项就是缺省的画直方图的形式。

绘制面积图需要用到函数 area。它的输入参数为矩阵或向量。该函数画出向量或矩阵的列向量中的数据，并连成线，填充横轴与曲线之间的面积。

4.4.4.3 矢量图

矢量图在工程技术中用得十分广泛，特别在场的理论中，引入矢量图使场的分布图形化，有助于理解场的结构，从而对场的各种特性有更深刻的理解。

在 MATLAB 中，提供了如下函数来绘制矢量图：compass，feather，quiver 和 quiver3。

这些函数如果只有一个参数，则把参数的实部当成 x 分量，把虚部当成 y 分量；如果有两个参数，则第一个参数为 x 分量，第二个参数为 y 分量。

函数 compass 绘制矢量时都是从同一个起点开始的，而函数 feather 则指定矢量的起点为横轴上等间距的点。调用函数 quiver 和 guiver3 时要指定每个矢量的起始点的坐标。

下面是一个关于物理的简单问题。在这里只是作为一个例子，并不涉及到高深的公式推导。

假定导弹沿 x 方向的速度为 vx，沿 y 方向的速度为 vy，且在水平方向上为匀速运动；沿 z 方向作匀加速运动，初速度为 vz，加速度为 a，初始位置为(0，0，h)，用矢量图画出导弹在空中各点的速度矢量。

```
%s224.m
vz=10;vx=2;vy=3;
a= 32;t=0:0.1:1;
x=vx*t; %x 坐标
y=vy*t; %y 坐标
z=vz*t+1/2*a*t.^2; %z 坐标
u=gradient(x); %速度的 x 分量
v=gradient(y); %速度的 y 分量
w=gradient(z); %速度的 z 分量
scale=0;
quiver3(x,y,z,u,v,w,scale); %绘制矢量图
axis square; %调整坐标系
set(gca,'Ydir','reverse');
```

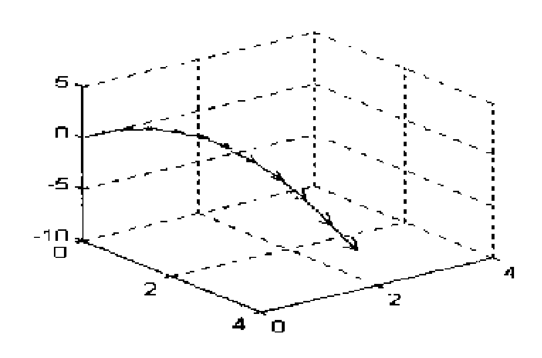


图 4-29 速度矢量图

绘制矢量图如图 4-29 所示。

4.4.5 制作动画

MATLAB 提供了以循环方式来查看图片的功能，熟练使用循环与观察点或其他命令可以作出许多动画片效果。

如下操作可看到图形按顺时针旋转。

```
%s225.m
z=peaks(50);
mesh(z);
hidden off;
deg=30;
[az el]=view;
```

```
rotvec=0:1:deg;
for i=1:length(rotvec)
   view([az+rotvec(i) el])
   drawnow
end
```

如下操作可观察到观察点往上移动的效果。

```
%s226.m
z=peaks(50);
mesh(z);
hidden off;
set(gca,'CameraFositionMode','manual');
index=get(gca,'CameraPosition');
dist=100;
for i=10:dist
   newposi=index-i;
   set(gca,'CameraPosition',newposi)
   drawnow
end
```

4.5 MATLAB编程

4.5.1 脚本与函数

4.5.1.1 脚本

脚本是M文件里的简单函数和命令的集合，它们没有输入输出参数，类似与DOS中的批处理文件。脚本可以在MATLAB环境下直接运行，它们可以访问工作空间的所有数据，由脚本建立的变量在脚本运行完后应存在于工作空间，可以对这些变量继续操作。

在图4-30所示的编辑窗口中输入。

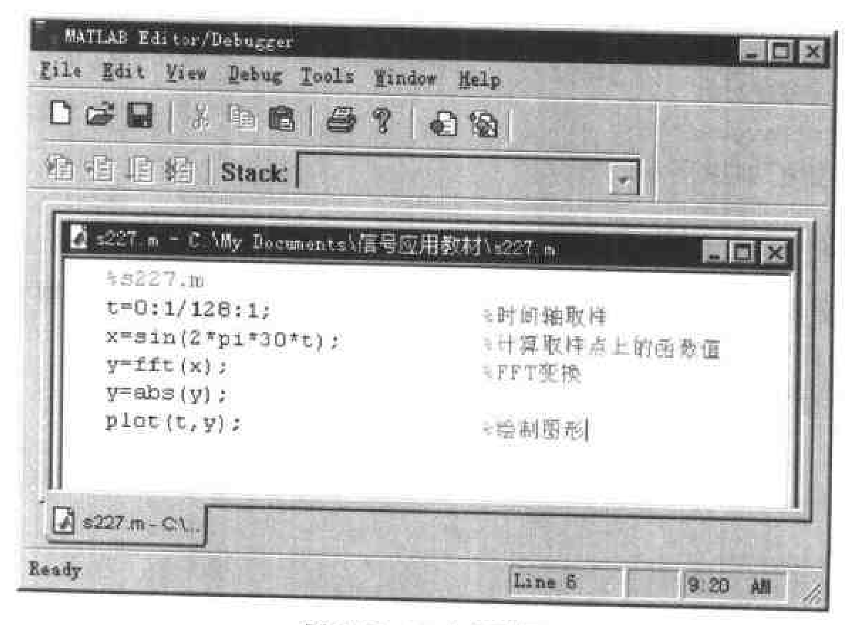


图4-30 M文件编辑

保存这一脚本，将其存为 s227.m 文件，然后在 MATLAB 命令窗口中键入文件名 s227，则在 MATLAB 的图形窗口会出现如图 4-31 所示的图形。

注意：①脚本程序实际上只包括两部分，即注释和指令，注释部分开头必须用%注明。②执行时可在命令窗口直接输入脚本程序名，也可用鼠标打开 File 菜单中的 Run Script 窗口再输入文件名。

图 4-31　FFT 变换曲线

4.5.1.2　函数

函数的使用与其他高级语言相似，MATLAB 提供的各种工具几乎都是以函数的形式给出的。函数接受输入参数，返回输出参数。它只能调用自己工作空间内的变量，MATLAB 命令窗口中的工作空间和其他函数的工作空间内的变量无法调用。

函数表示如图 4-32 所示。

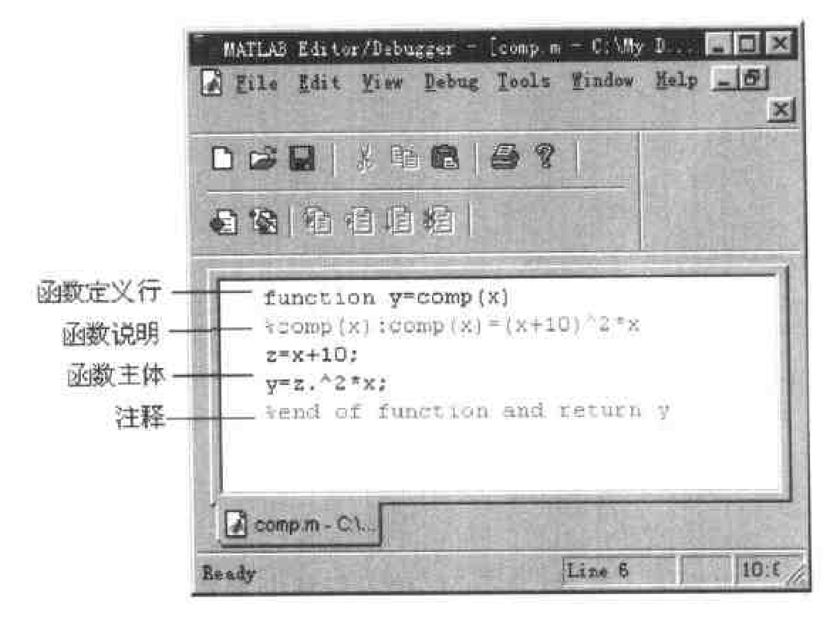


图 4-32　函数的表示

函数主要包含如下四种成分：

✧ 函数定义行：要求用关键字“function”把该文件定义为函数，并指定函数的名字，该名字必须与存盘的文件名相同。

✧ 函数主体：完成函数的主要功能，包括运算、赋值操作、作图等。

✧ 函数说明：对本函数完成功能的简要说明。

✧ 注释：是夹杂在函数体中的语句，它以%开头，出现%的行，%后的语句均为注释性的。

上述四项除了函数定义行必须有，另外三部分可以没有，如空函数。

保存函数 comp.m 文件，要执行这个函数，可以在命令行上直接输入

```
comp（20）;
```

或 k=20;

```
comp(k);
```

或 m=comp(20);
此函数完成 $y=(x+10)^2 x$ 的运算。

4.5.1.3 函数的参数传递

函数有自己的工作空间，它与 MATLAB 的工作空间分开。因此，函数内的变量与 MATLAB 工作变间的联系必须利用输入输出参数。

函数空间有两个固定的变量“nargin”和“nargout”，分别等于实际的输入参数个数和输出参数个数。函数 nargchk 用于检变输入参数个数是否正确，调用格式为：

```
msg=nargchk(low,high,n)
```

该函数作用是检查 n 是否在 low 和 high 这两个数中间，若不在返回错误信息。利用上面两个变量和一个函数可以对输入参数、输出参数的情况进行判断，并及时给出信息。

利用 nargin 可以判断出输入参数的个数，从而决定采取什么操作。如定义一个函数 g，它可以接变一个参数，此时对该参数求其相反数；也可以接受两个参数，此时求两个数的和；接受三个参数，此时求第一个参数与第三个参数的差。

```
function[t,r]=g(varargin)
switch nargin
case 1
  r=-1*varargin{1};
  t='good';
case 2
  r=varargin{1}+varargin{2};
  t='sum';
case 3
  r=varargin{1}-varargin{3};
  t='minus';
 otherwise
  disp('error');
end
```

这里用 varargin 作为输入参数，它可以传入任意个参数给函数，调用参数的格式是 varargin{1}、varargin{2}、…。同样可以用 varargout 来输出任意个参数，方法相同。语句 switch nargin 确定输入参数个数，以决定作何种操作。在 MATLAB 命令窗口中调用 g 函数的结果如下。

```
[t,r]=g(3) %传递给 g 函数一个参数
t =good
r = -3
[t,r]=g(2,3) %传递给 g 函数两个参数
t =sum
r = 5
[t,r]=g(1,2,3) %传递给 g 函数三个参数
t =minus
r =-2
[t,r]=g(1,2,3,4) %传递给 g 函数四个参数
error
Warning: One or more output arguments not assigned during call to 'g'.
```

4.5.1.4 P代码

除了.m 文件之外，MATLAB 提供了另一种格式码称 P 代码。任何.m 文件都可通过下列命令保存为 P 代码文件：

pcode filename.m

如在命令行输入：

pcode ex2

在同一目录下就会生成一个 ex2.p 文件，P 代码可以和 M 文件一样执行。如果目录下同时有 P 代码和 M 文件，MATLAB 会执行 M 文件。

P 代码的优点：

① 保密性：P 代码的内容为伪码，无法向文本文件一样查看其内容。

② 快速性：在建立 P 代码的同时，MATLAB 会对 M 文件程序进行分析，在下一次执行 P 代码时，可减少一个分析步骤。对需要进行大量图形处理的情况，就会大大提高效率。

4.5.2 MATLAB 编程的基本知识

4.5.2.1 变量类型

（1）MATLAB 对变量名称的规定

① 变量名称的开头必须是一个英文字母，后面接的可以是英文字母、数字和下划线。

② 区分大小写。

③ 变量名称不能超过 31 个字符。

（2）全局变量和固定变量

局部变量：变量只在一个函数内部使用，不用特别声明。

全局变量：两个或多个函数共用一个变量，用 Global 来声明。

固定变量：在程序中固定不变的量。

如：

ans:最近运算的结果，它用于自动存储未制定输出变量的表达式的结果。

eps:浮点运算的相对精度，即 MATLAB 运算的误差。

pi:圆周率 3.1415926…。

j,i:虚数单位。

inf:无穷大。

NaN:无效的数值。

computer：用户计算机的类型。

flops：计算到当前为止系统执行基本运算的次数。

version：MATLAB 的版本号。

realmax：用户计算机能表示的最大的正浮点数值。

realmin：用户计算机能表示的最小的正浮点数值。

4.5.2.2 数据类型

MATLAB 定义了六种数据类型，即

char：字符数组，每一个字符用 16 位来表示。

如：char 'Luoson–1'

double：双精度值数组，是 MATLAB 中最常用的数据类型。

如：double [2.5,4;0.3,0]

sparse：双精度值稀疏数组。

如：sparse speye(4)

unit8：无符号的 8 位整数数组，数字范围为 0–255。

如：Unit8(magic(3))

cell：单元数组，可以包含其他不同类型数据的数组，适合大型数据库使用。

如：cell {'Luoson-1' 100 eye(2)}。

struct：结构数组，与单元数组一样，可以将不同的数据类型包含在同一变量名称下。

如：struct Cat.name='mimi'
Cat.age=1.4
Cat.color='yellow'

4.5.2.3 流程控制

MATLAB 中提供了条件分支语句、循环语句等常用的控制语句，使得 MATLAB 编程相当容易。

（1）条件分支语句

条件分支语句有 if 语句和 switch 语句两种。

```
if 语句格式如下：
if（逻辑表达式）
    语句 1
elseif（逻辑表达式）
    语句 2
    ……
else
    语句 n
end
```

示例如下。

```
%s228.m
x=6;
if((x+3)<2) %利用条件判断语句判断 x 值
   y=x; %决定程序运行流程
elseif(x<4)
   y=2*x;
else
   y=x.^2;
end
```

执行结果 y=36

switch 语句格式如下。

switch 表达式（标量或字符串）
case 值 1
语句 1
case 值 2
语句 2
……
otherwise
语句 n
end

示例如下。

```
%s229.m
x=2;
switch(x) %如果 x 为 1
case  1
   y=x
case  2 %如果 x 为 2
   y=x*2
otherwise %如果不是以上情况
   y=x-1
end
```

执行结果 y =4

（2）循环语句

循环语句有 for 和 while 两种。

for 语句格式如下：

for 循环变量=起始值：步长：终止值
循环体语句
end

例如：

```
%s230.m
y=0;
for i=1:2:100
   y=y+i;
      end
```

执行结果 y=2500

for 循环也可以嵌套进行，如：

```
for i=1:5
   for j=1:4
      y(i,j)=i*j;
   end
end
```

运行结果如下。

```
y = 1    2    3    4
    2    4    6    8
    3    6    9    12
    4    8    12   16
    5    10   15   20
```

多层循环时也应当有多个 end 标志，个数与循环层数相同。

while 语句格式如下：

```
while 循环变量=起始值：步长：终止值
      循环体语句
end
```

4.6　MATLAB 的接口

MATLAB 是一个开放的系统，它具有多种接口功能，使得用户可以十分方便地与其他的应用程序交换数据和信息。它的接口主要有：数据输入输出接口、子程序库调用接口 、图形图像输入输出接口、动态链接接口等。

4.6.1　MATLAB 的数据接口

MATLAB 接受两种形式的数据：①ASCII 码的文本数据文件；②MATLAB 定义的 MAT 型数据，即所谓的 MAT 文件。

4.6.1.1　*数据结构*

了解 MATLAB 的内部数据结构和数据对象，对于进行数据交换操作是十分必要的。数据结构包含以下意义：变量名 NAME、变量的维数 Dimensions、变量的类型 ClassName、变量的实数或复数类型、变量的存储属性 Storage。例如输入：A=[1 2;3 4;5 6]，那么矩阵 A 的属性如下：

NAME = “A”

Dimensions=3×2

ClassName=double

Storage=FULL

4.6.1.2　MATLAB *的数据输入*

用户可以用多种方式向 MATLAB 系统输入数据，最佳的输入方式依赖于拥护的数据格式。下面给出几种常用的方法。

（1）显示的输入

数据量比较小的情形，可以在 MATLAB 命令窗口中，从键盘直接输入待输入的数据。即使用方括号将矩阵的元素按行的顺序括起来，每行用分号隔开，行内各元素用空格或逗号分开。

（2）M 文件形式输入

如果数据量较大，且数据是某种硬拷贝形式，不是以计算机可读形式存在时，最有效的办法是，用某种文本编辑器直接键写一个包含数据矩阵的 M 文件。

（3）ASCII 码数据文件的输入

MATLAB 可以直接读入 ASCII 码的数据文件。ASCII 码的数据文件中的数据形式必须是一个矩阵，要求数据文件每一行的数据个数必须相同，每行数据对应于矩阵的每一行，每行的元素用空格分开。用户可以将数据编辑成一个 ASCII 码文件，或者由其他的程序将所输出的数据写到一个 ASCII 码文件中。ASCII 码文件中的数据由 MATLAB 的命令 load 装入，命令形式是：

>>load 文件名(带文件扩展名)

该语句在 MATLAB 工作空间中创建一个与文件名(无文件扩展名)相同的变量，该变量表示的矩阵即是 ASCII 码文件的数据组成的矩阵。

（4）低层 I/O 输入

MATLAB 提供了文件低层操作函数，可以直接打开文件(fopen)，读文件(fread)。这种方法主要用于装入某种特定格式的数据文件，这种数据文件可能是由其他的应用程序生成和创建的。

（5）MEX 动态程序输入

如果已经有一些子程序(如 C 子程序或 FORTRAN 子程序)可以用来读取某些特定格式的数据文件，那么可以开发 MATLAB 的动态链接 MEX 子程序，与已有的子程序链接在一起，将数据文件转换成 MATLAB 的 MAT 数据文件，再用 load 函数将有关的数据装入到 MATLAB 系统中。

（6）外部程序转换

如果已有的数据文件格式比较复杂，用户可以开发 FORTRAN 或 C 程序将数据文件直接转换成 MATLAB 的 MAT 数据文件，再用 load 命令装入到 MATLAB 系统中。在这种情况下，必须对 MAT 文件结构有较深入的了解。

4.6.1.3 MATLAB 的数据输出

MATLAB 系统输出数据的方式主要有以下几种。

（1）小型矩阵输出

对于数据量较小的数据矩阵，通过输出的设置，可以用 diary 命令生成命令窗口部分内容的拷贝文件。该文件是文本文件，记录了命令窗口的命令行，以及 MATLAB 的屏幕输出内容。这样可以将 diary 文件剪接到其他的文件或报告中。

（2）ASCII 码数据输出

带-ascii 选项的 save 命令，可以生成一个 ASCII 码的数据文件。在这种方法中，用户可以存储某些指定的变量。例如，下列语句

```
>>A=rand(4,3)
>>save temp.dat A -ascii
```

将生成名字为 temp.dat 的 ASCII 码文件，包含矩阵 A 的全部数据，如

```
0.9501   0.8913   0.8214
0.2311   0.7621   0.4447
0.6068   0.4565   0.6154
0.4860   0.0185   0.7919
```

（3）低层 I / O 输出

利用 MATLAB 的低层 I / O 函数 fopen 和 fwrite，或者其他低层 I / O 函数，可将数据写入到某个特定格式的文件中。这种方式主要用于将输出的数据写成某些其他应用程序所需要的数据格式。

（4）MEX程序输出

如果已经有某些子程序可以将数据写成特定格式，那么可以开发MEX子程序直接调用那些子程序，把最后形成的数据格式作为某些应用程序的输入数据。

（5）MAT格式输出

利用MATLAB的save命令，可以将数据存储成MAT格式文件，再开发一些C或FORTRAN程序将MAT文件转换成某些所需要的特殊格式。例如，将MATLAB的图像数据存入到MAT文件中，然后利用C或FORTRAN程序将MAT的图像数据转换成一些标准的图像格式文件，如PCX、TIF、GIF格式等。

4.6.1.4 MATLAB的数据格式

MAT数据格式是MATLAB的数据存储的标准格式。一个MAT文件中可以存储一个或多个矩阵数据，矩阵顺序地存在一片连续的磁盘空间上。在每个矩阵的开始处，有一个固定长度的矩阵信息头。这个信息头完整地描述该矩阵的全部特征信息，信息头之后才是矩阵的数据部分，数据占用的磁盘空间字节长度由信息头的长度信息决定。整个这样的结构就构成MAT文件中的一个矩阵。

MATLAB的save命令可以将MATLAB系统内部数据写为MAT文件，而load命令可以将磁盘上的MAT文件正确地读入到MATLAB系统中。除此之外，为了有效地管理MAT文件，以及在MATLAB外部读取和创建MAT文件，MATLAB提供了一个子程序库，用户可以在C或FORTRAN程序中直接调用这些子程序来创建和读取MAT文件。如果用户了解MAT文件的结构，则完全可以开发自己的子程序读写MAT文件。

4.6.2 文件I/O操作

MATLAB系统具有直接对磁盘文件进行访问的功能。这样不仅可以进行高层的程序设计，也可以进行低层次磁盘文件的读、写操作，极大地增强了MATLAB程序设计的灵活性。

4.6.2.1 文件的打开与关闭

根据操作系统的要求，在程序要使用或创建一个磁盘文件时，必须向操作系统发出打开文件的命令。文件使用完毕后，还必须告诉操作系统关闭使用过的文件。

（1）打开文件

在读写一个ASCII或二进制文件前，必须先执行打开文件的操作：

fid=fopen('filename','permission')

参数permission的选择：

r：表示对文件进行数据“只读”操作；

w：表示对文件进行数据“只写”操作；

a：表示对文件进行数据“追加”操作，在文件尾追加数据；

r+：表示对文件进行数据“读写”操作；

w+：表示创建一个新文件或删除已存在的文件内容，并进行数据“读写”操作；

a+：表示创建一个新文件，打开一个已存在的文件，并进行数据“追加”操作。

使用方法如下：fid=fopen('first.dat','r')

用函数fopen以只读的形式打开磁盘文件first.dat，用fid句柄标识打开的文件。

（2）关闭文件

当完成了打开文件或写文件的操作后，在要结束前必须做关闭文件的操作。

使用方法如下：status=fclose(fid)

用函数 fclose 即可关闭句柄为 fid 的已打开的文件，用 status 句柄标识关闭的文件。

4.6.2.2 二进制数据文件的读写操作

（1）读文件操作

用 fread 来读二进制数据文件，例如

```
fid=fopen('first.dat','r');
A=fread(fid);
status=fclose(fid);
```

或 A=fread(fid，100)；读文件中前 100 个数据到列向量 A 中。

或 A=fread(fid，Inf)；读至文件结尾，生成列向量 A。

（2）写文件操作

用 fwrite 将一个矩阵的元素按一定的数据精度写入到某个打开的文件中，例如

```
fwriteid=fopen('first1.dat','w');
count=fwrite(fwriteid);
status=fclose(fwriteid);
```

4.6.2.3 文件内的位置控制

根据操作系统的规定，在读写数据时，缺省的方式总是从磁盘文件的开始处顺序向后地在磁盘空间上读写数据。操作系统控制一个文件指针，指出当前文件位置。MATLAB 的函数 fseek 和 ftell 可以用来设置文件指针的位置，并取得文件指针的当前位置。该位置是下一次读写操作的起始点。

fseek：设定文件位置指示器的位置。

ftell：返回文件位置指示器的位置。

bof：文件头。

eof：文件尾。

如

```
fid=fopen('first.dat','r');
st=fseek(fid,9000, 'bof');
s=fread(fid,100,'int16');
s1=fclose(s);
```

从打开的二进制文件 first.dat 中的前数第 9000 位的数据开始，读 100 个数据到变量 *s* 中，并关闭打开的文件。

4.6.2.4 格式文件的输入和输出

输入或输出特定格式的文件。

（1）格式输入

用 fscanf 输入一行特定格式的数据，将所需的数据按参数的类型放在变量中，参数以“%”

开头，常用参数有：

%s　字符串

%d　十进制的整数

%g　浮点数

如：

```
fid=fopen('first.dat','r');
s=fscanf(fid,'%d');
status=fclose(fid);
```

从 first.dat 中内容读取到变量 s 中，内容是整数值。

（2）格式输出

用 fprintf 将数据转换成字符串，输出到命令窗口或写入到一个文件中。输出数据的格式说明参数以“%”开头，常用参数有：

%e 数据指数表示形式

%f 固定小数点位置的数据格式

%g 在和两种格式中自动选取较短的格式

如：

```
x=0:0.1:1;
y=[x;exp(x)];
fid=fopen('exptable.txt','w');
fprintf(fid,'Exponential Functuion\n\n');
fprintf(fid,'%6.2f   %12.8f\n',y)
status=fclose(fid);
```

这个例子创建一个文本文件 exptable.txt、包含指数函数的函数值表。第一条 fprintf 语句输出一行标题，随后空两行；第二条 fprintf 语句输出函数值表自身，每组自变量值和函数值占一行，都是固定小数点位置的格式。自变量值占 6 个字符位，而其小数点后的精度是 2 个字符位；函数值占 12 个字符位，而其小数点后的精度是 8 个字符位。自变量值和函数值之间空两格。

矩阵 y 的元素按列的顺序转换成格式化的输出，函数反复使用格式说明，直到将矩阵的数据全部转换完毕。

4.7 Simulink

Simulink 是实现动态系统建模、仿真的一个集成环境。它的存在使 MATLAB 的功能得到进一步的扩展。这种扩展的意义表现在：①实现了可视化建模，用户通过简单的鼠标操作值可建立起直观的系统模型，并进行仿真；②实现了多工作环境间文件互用和数据交换，如 Simulink 与 MATLAB\FORTRAN 以及 C 的工作环境的信息交换都可以方便地实现；③把理论研究和工程实现有机地结合在一起。

在 MATIAB 电子仿真中，将会大量地用到 Simulink，所以本节只介绍关于 Simulink 基础知识，更为具体的使用方法，将在应用中通过例子来详细介绍。

4.7.1 Simulink 简介

近几年，在学术界和工业领域，Simulink 已经成为在动态系统建模和仿真方面应用最

广泛的软件包之一。它的魅力在于强大的功能和简便的操作。作为 MATIAB 的重要组成部分，Simulink 具有相对独立的功能和使用方法。它是对动态系统进行建模、仿真和分析的一个软件包。它支持线性和非线性系统、连续时间系统、离散时间系统等，而且系统可以是多进程的。

Simulink 为用户提供了用方框图进行建模的图形接口，采用这种方法进行系统设计，就像你用笔和纸来画一样容易。它与传统的仿真软件包用微分方程和差分方程建模相比，具有更直观、方便、灵活的优点。

需要提出的一点是，用 Simulink 创建的模型可以具有递阶层次结构，及允许用户建立自己的子系统。在查看时，用户可以从最顶层开始，然后用鼠标双击其中的子系统模块，从而进入子系统模块进行查看，这样非常便于模型的条理化，从而帮助用户理解模型的整体结构以及各模块之间的关系。

4.7.2 一个使用 Simulink 的例子

为了建立对 Simulink 的基本认识，先介绍一个例子。在这个例子中，用 Simulink 来实现两个正弦信号的相乘，即计算 $x(t)=\sin t\times\sin(10t)$。

第一步：运行 Simulink

在 MATLAB 命令窗口中运行 Simulink 有三种方式：

① 在 MATLAB 命令窗口中直接键入“Simulink”；

② 单击 MATLAB 工具条上的 Simulink 图标“ ”；

③ 在 MATLAB 菜单上选择 File|New|Model。

运行后将会显示如图 4-33 左图所示的“Simulink 模块库浏览器”，如果采用上面的方式③，还将看到如图 4-33 右图所示的一个新建模型窗口。如果只打开了“Simulink 模块浏览器”，可以通过单击该窗口工具栏上的“新建”图标，来得到一个新建模型窗口。

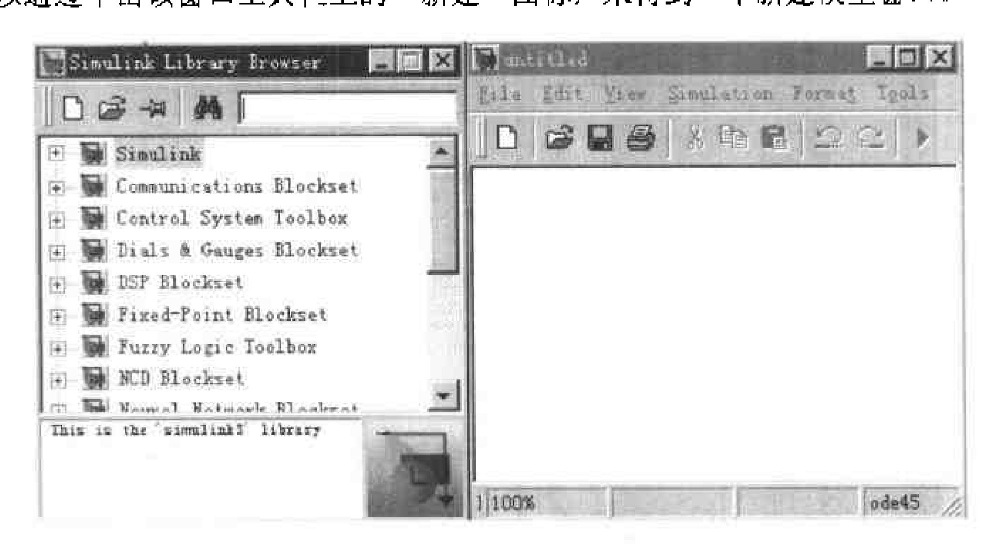


图 4-33 Simulink 界面

第二步：将所需模块添加到模型中

模块浏览器非常类似 Windows 的资源管理器，它的作用是将各种模块组织在不同的目录中，以便用户选取。从模块浏览器可以直接看出，位于顶层的有：

① Simulink: Simulink 基本模块库；

② Communications Blockset:通信模块库；

③ DSP Blockset:数字信号处理模块库；

④ Control System Toolbox:控制系统模块库。

分别单击模块浏览器中 Simulink、 Communications Blockset、 DSP Blockset 前面的“+”，看到各个库中包含的子模块库，如图 4-34 所示。

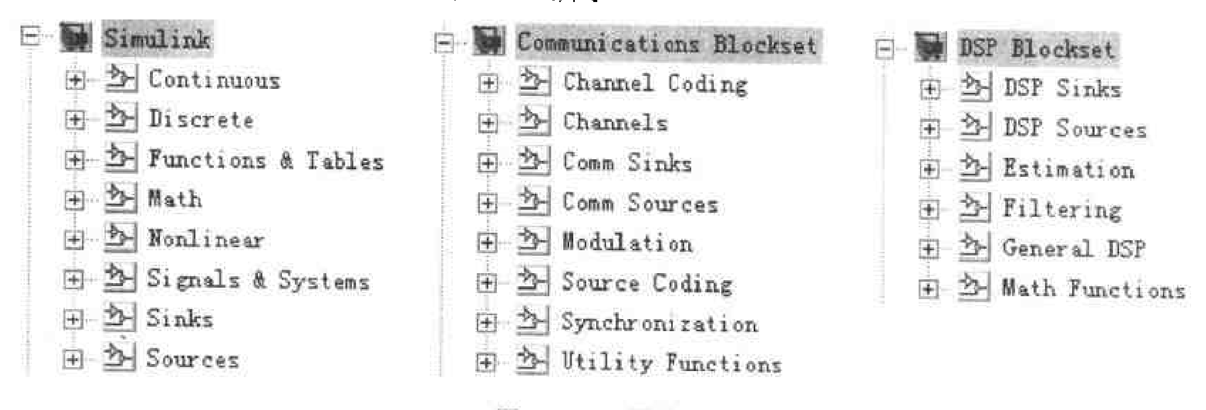


图 4-34　子模块库

将本例中需要的三种模块——正弦源、相乘器和示波器添加到新建模型窗口中。首先点击“Sources”前面的“+”号，将看到输入源子模块库中的函数，我们需要的是 Sine Wave(正弦源)。直接用鼠标点中正弦源，然后不要松开鼠标，将其拖入模型窗口中后再松开鼠标，这样会在模型窗口中看见正弦源的方框图。由于本例中需要两个正弦源，依此方法再拖入第二个正弦源。

照此方法，通过单击 Sinks 前面的“+”号，将 Sinks(接收)子模块库打开，接着把 Scope(示波器)拖入模型窗口中。接着把 Math(数学)子模块库中的 Product(相乘器)拖入模型窗口中。最后看到的结果如图 4-35 所示。

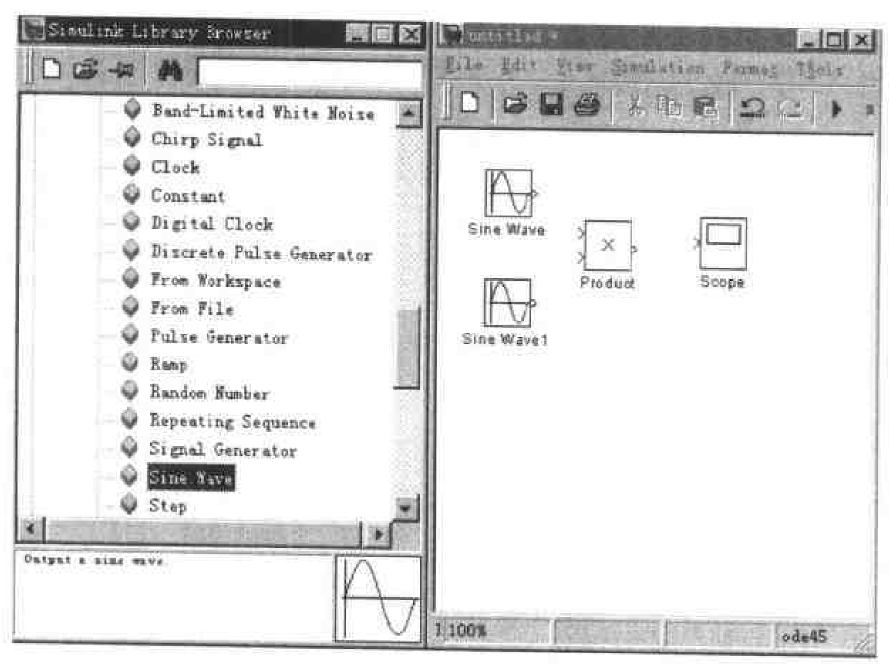


图 4-35　将模块加到模型中

第三步：编辑模块组成模型

将模块添加到模型中以后，还要编辑模块的性质以满足要求。两个正弦源的频率分别是

1Hz和10Hz，幅度均为1；还要求显示两个正弦波以及相乘后的波形，因此要求示波器有三个输入端。

只要双击一个模块就可以修改该模块的参数(Parameters)。首先修改两个正弦源的参数。先双击某一个正弦源，可以看到如图4-36所示的对话框。

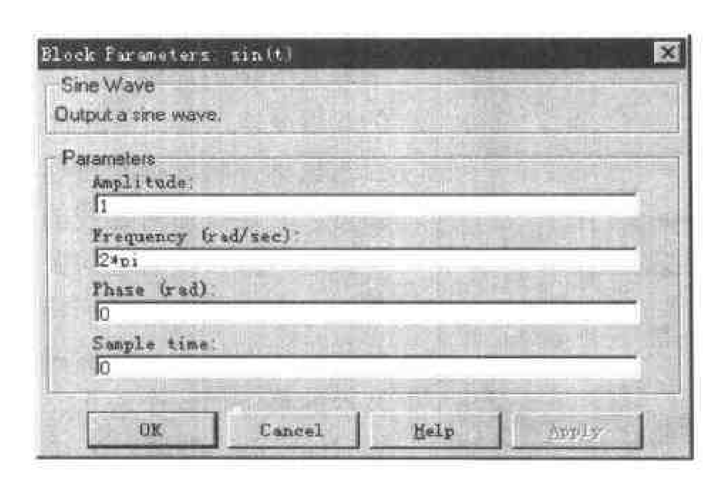


图4-36 修改正弦源的参数

框分为两部分，上面一部分描述了该框图的作用：输出一个正弦波；下面一部分是可以调整的参数，包括4项：幅度、频率、相位和采样时间。注意到这里的频率是角频率，将其改为2*pi，其他参数则不用修改。在模型中，可以更改某个模块的名字。现在看到的正弦源的名字是默认值：Sine Wave，可以单击这个名字，然后将看到一个文本框，可以在其中输入新的模块名称，本例中输入sint。这样我们就完成了对一个正弦源的编辑。

对第二个正弦源，我们将其参数中的角频率改为“20*pi”，名称改为sin(10t)。

对于相乘器(Product)，通过双击该模块可以发现，它只有一个参数——输入信号的个数，我们采用默认值2。

当双击示波器时，将看到一个图形界面（图4-37左图所示）。单击工具栏上的图标，将出现示波器属性设置窗口（图4-37右图所示）。

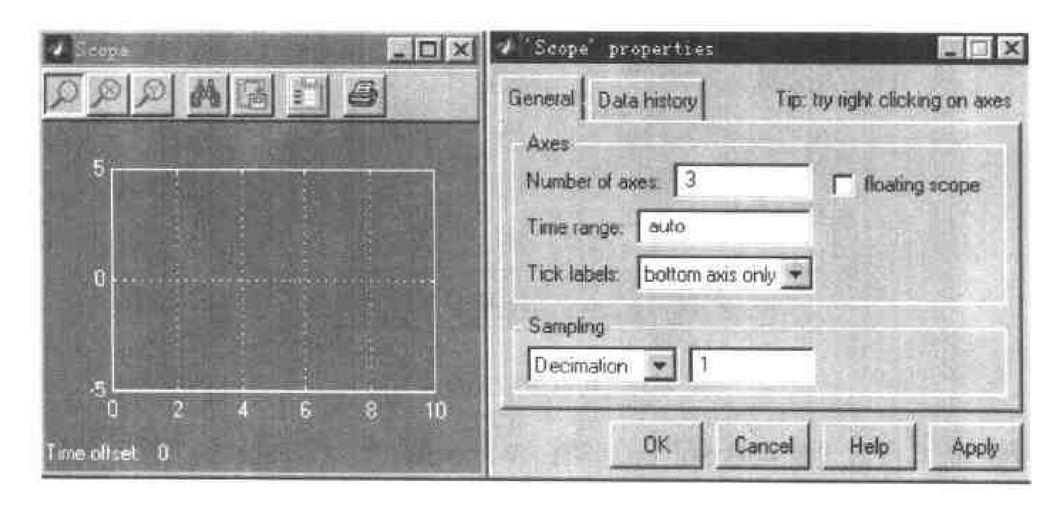


图4-37 修改示波器的属性

“Number of axes”改成3，这表明需要用这个示波器观察三个波形。当示波器改成三个输入端口后，这个框图显得有点拥挤，其实我们可以改变它的大小。先单击选中该示波器，

我们可以看到这个模块周围有四个黑点，用鼠标点住其中任何一个，拖动改变其大小，直到令人满意为止。

***小技巧*：有时为了连线方便，需要将模块的端口挪到合适的位置，可以用“Ctrl+R”来实现将某个模块顺时针旋转 90°，用“Ctrl+F”实现将某个模块翻转 180°。**

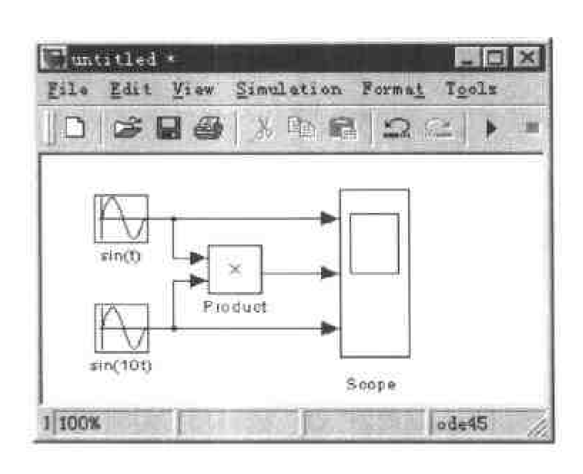


图 4-38　实现信号相乘的系统图

最后将整个模型连接起来。Simulink 提供了非常方便的连线方法。当将鼠标置于某模块的输出或输入端口附近时，鼠标将变成十字形，点击鼠标并将其拖至待连接的另一模块的端口即可。若连线后对连线的走向不满意，可以选中该线，然后拖着它的关键点移动，直到满意为止。在本题中，我们将两个正弦源的输出连至相乘器的输入，然后将相乘器的输出接到示波器的最下面一个输入。下面我们想把正弦源同时输出到示波器，这时就要用“Ctrl”加上鼠标点击原连线，然后再拖到目的地，这样就可以实现在一条连线上分叉。最后实现的系统如图 4-38 所示。

第四步：进行系统仿真

前面已经将整个信号相乘的系统完整地搭建出来，剩下的就是进行仿真了。在这一步骤里，我们将设置仿真的时间、步长以及算法。这些参数都在 Simulink 菜单下的“Parameters”命令中设置。单击该命令，可以看到仿真参数设置对话框，如图 4-39 所示。从这个对话框

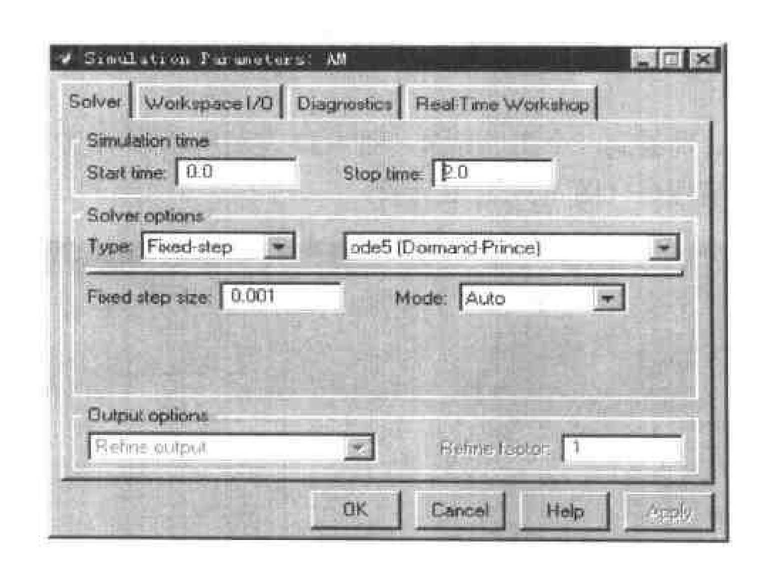


图 4-39　仿真参数设置对话框

中可以看到，仿真参数设置包括 Solver(解法)、Workspace I / O Page(到工作区的输入输出)、Diagnostics(诊断)和 Real_Time Workshop(实时工作间)。针对本例，介绍 Solver 的设置方法，其余部分将在应用中介绍。

可以看到，整个 Solver 选项卡包括三个部分：

（1）设置起始时间和停止时间

选项的最上面是仿真时间设置，以秒为单位给出仿真始末时间，在本例中为 0 到 2s。

（2）设置仿真步长和算法

选项的中间部分是解法选项，包括步长设置和算法设置。在 Type 选项中，步长可以设为 Variable—step(不定步长)和 Fixed—step(定步长)，本例选择定步长计算，并将最大步长设为 0.001s。步长设置得越小，计算越准确，绘图越圆滑。如果选择“auto”，则采用默认步长，为仿真时间的 1/50。算法设置有很多可选项，一般对于离散系统，要选择 discrete 算法，而对于连续系统，选择 ode 系列算法。本例中选择 ode5，即基于龙格-库塔(Runge-Kutta)法的 5 阶算法。一般算法采用的阶数越高，计算越准确。

（3）设置输出选项

选项的最下择分是输出设置，仅在不定步长仿真时有效。有三种选择：细化输出、产生附加输出和只产生指定输出。细化输出的值越大，输出曲线越光滑。本例由于采用定步长算法,不能设置此项。

到现在为止,已经完成了整个模型的设计工作。在执行仿真之前，先将这个模型保存，单击模型窗口的“保存”图标，或 File 菜单下的 Save 命令，将这个模型命名为“AM”(因为这个系统实际上就是通信中的抑制载波调幅)保存在 MATLAB 下的 work 目录下，这个目录在每次 MATLAB 启动时都被设为工作目录，从而保证了这个模型随时可以被引用。

通过单击模型窗口中的▶图标，或从 Simulink 菜单中运行 Start 命令来运行仿真。双击示波器，可以看到仿真的结果如图 4-40 所示。

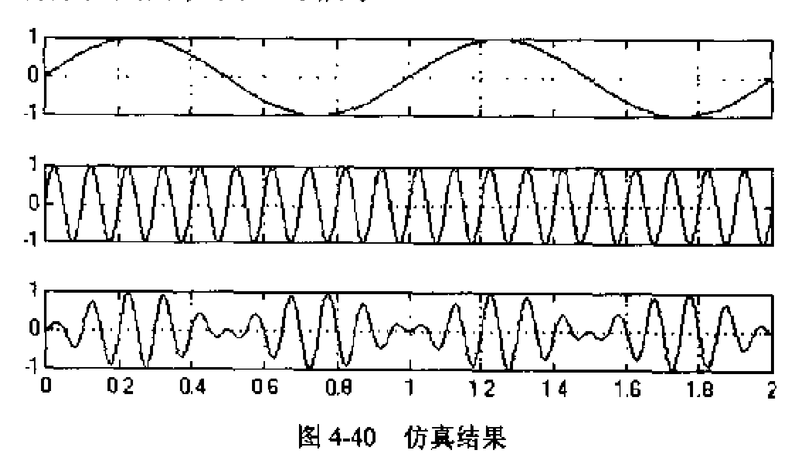


图 4-40 仿真结果

可能在自己的计算机上看到的图形与图 4-40 不一样，主要是因为坐标刻度设的不一样。我们在某一张波形图上右击，选择菜单中的“Axes Properties”，可以修改纵坐标的范围，也可以给波形加上一个标题。有时我们直被单击示波器工具栏上的■图标，可以自动缩放坐标来使波形刚好完整显示。

至此就完成了这个例子。通过这个例子。对 Simulink 的使用已经有了一个新的认识。关于 Simulink 环境的内容实在太多了，有关 Simulink 的知识需在应用中慢慢体会。

4.7.3 Simulink 模块的模作

4.7.3.1 添加和选取模块

（1）添加模块

在 Simulink 模块库中找到要添加的模块，用鼠标将这个模块拖入模型窗口。

（2）选取模块

当模块已处于模型窗口中时，只要用鼠标在模块上单击就选中该模块。

4.7.3.2 模块位置和外形的调整

（1）调整模块位置

用鼠标左键点住模块，按住左键拖动它到适当位置，松开鼠标。

（2）调整模块大小

模块被选中后，会看到模块四角的关键点，用鼠标点住模块的关键点，按住拖动它到适当大小，松开鼠标。

（3）调整模块方向

"Ctrl+R"将某个模块顺时针旋转 90°，或使用 Format 菜单选择 Rotate Block。

"Ctrl+F"将某个模块翻转 180°，或使用 Format 菜单选择 Flip Block。

（4）调整模块颜色和效果

选中模块后，使用 Format 菜单选择 Foreground Color，在弹出的菜单中选择模块的背景颜色：模块的图标、边框和模块名的颜色。

选中模块后，使用 Format 菜单选择 Background Color，在弹出的菜单中选择模块的背景颜色：模块的背景填充色。

选中模块后，使用 Format 菜单选择 Show Drop Shadow，调整模块的效果。

4.7.3.3 模块名的处理

（1）改变模块名

在 Format 菜单下，选取 Font 命令。

（2）隐藏模块名

在 Format 菜单下，选取 Hide Name 命令。

（3）改变模块名的位置

在 Format 菜单下，选取 Flip Name 命令。

4.7.3.4 复制和删除模块

（1）复制模块

① 模住 Ctrl 键不放，用鼠标左键点住要复制的模块，模住左键拖动该模块，放到适当的位置松开鼠标。

② 使用 Edit 菜单下的 Copy 和 Paste 命令。

（2）删除模块

① 使用 Edit 菜单下的 Cut 命令将模块剪切到剪贴板。

② 使用 Edit 菜单下的 Clear 命令将彻底删除模块。

③ 使用 Delete 键彻底删除模块。

4.7.3.5 模块参数和属性的设置

（1）模块参数的设置

① 使用鼠标双击一个模块就可以打开参数设置对话框。

② 使用 Simulation 菜单下的 Parameters 命令，也可以打开参数设置对话框。

（2）模块属性的设置

使用 Edit 菜单下的 Block Properties 命令，得到属性设置对话框。

属性的意义：

Description：对该模块在模型中的用法进行说明。

Priority：规定该模块在模型中相对其他模块执行的优先顺序。该数值越小，优先级越高，数值必须是整数。

Tag：用户模块添加的文本格式的标记。

Open function：当双击该模块时调用的 MATLAB 函数。

Attributes function string：用来在模块旁边显示字符串。这里的内容将在模块旁边显示出来，但不是模块名的一部分。

4.7.3.6 模块间的连接

（1）连接模块

移动鼠标到输出端，鼠标的箭头会变成十字形光标，点住鼠标左键，移动鼠标到另一模块的输入端，当十字形光标出现“重影”时，松开鼠标即可。

（2）模块间连线的调整

用鼠标单击连线，可以选中连线，这时会看到线上的关键点。用鼠标点住关键点，拖动可改变连线的走向。

（3）连线的分支

选好一条线，把鼠标移到分支的起点位置，先拖 Ctrl，然后拖住鼠标，将连线拖到目标模块，松开 Ctrl 和鼠标即可。

（4）标注连线

双击某一连线，可以看到一个文本框，在里面输入标注文字，按鼠标左键确定。

（5）移动连线

把鼠标指到要移动的线段上，按住鼠标左键，鼠标的箭头会变成十字形，移动线段到目的地，松开鼠标左键。

（6）分割连线

把鼠标指到分割线段的控制点上，直到鼠标的箭头变“o”形．拖住鼠标左键，拖动线段可以分割连线，松开鼠标左键。

（7）分离模块

将鼠标指针放在想分离连线的模块上，按住 Shift，再用鼠标把模块拖到别处即可。

4.7.3.7 建立子系统

① 使用 Library 中的“Connections”模块函数库。

② 利用 Edit 菜单内的“Create subsystem”功能。

4.8 Simulink 基本模块介绍

运行 Simulink 后，单击模块浏览器中的图标，选择 MATLAB 目录下的 toolbox/simulink/blocks/simulink3 模型打开，将看到如图 4-41 所示的 Simulink3.0 版的模块库。

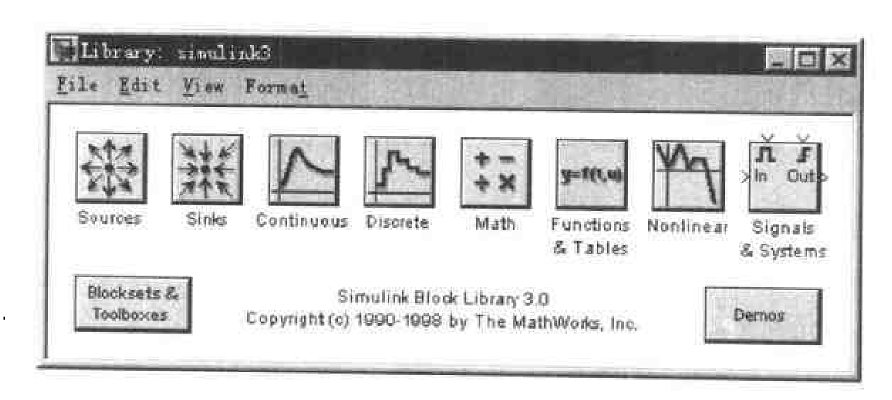


图 4-41 Simulink3.0 版模块库

4.8.1 输入源模块(Sources)

双击 Sources 模块库，看到输入源模块的种类如图 4-42 所示。表 4-5 介绍输入源模块的功能。

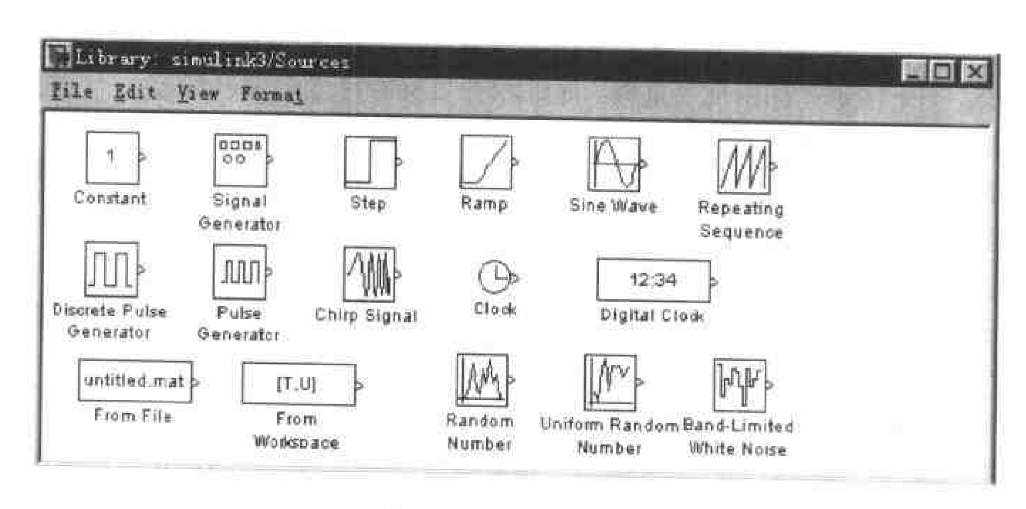


图 4-42 输入源模块

表 4-5 输入源模块简介

模 块 名	功 能 简 介	模 块 名	功 能 简 介
Constant	常数	Chirp Signal	频率不断变化的正弦信号
Signal Generator	信号发生器	Clock	输出当前的仿真时间
Step	阶跃信号	Digital Clock	按指定速率输出当前仿真时间
Ramp	线性增加或减小的信号	From File	从文件读数据
Sine Wave	正弦波	From Workspace	从当前工作空间定义的矩阵读数据
Repeating Sequence	重复的线性信号，类似矩齿形	Random Number	高斯分布的随机信号
Discrete Pulse Generator	离散脉冲发生器，和采样时间有关	Uniform Random Number	平均分布的随机信号
Pulse Generator	脉冲发生器，和采样时间无关	Band-Limited White Noise	带限白噪声

4.8.2 接收模块(Sinks)

双击 Sinks 模块库，看到接收源模块的种类如图 4-43 所示。表 4-6 介绍接收源模块的功能。

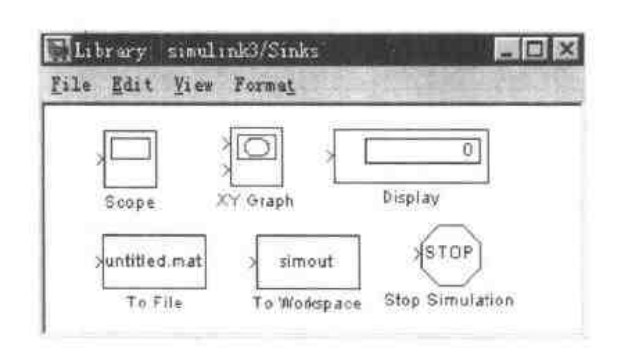


图 4-43 接收源模块

表 4-6 接收源模块简介

模 块 名	功 能 简 介	模 块 名	功 能 简 介
Scope	示波器	To File	保存到文件
X Y Graph	两个信号的关系图	To Workspace	输出到当前工作空间的变量
Display	实时数值显示	Stop Simulation	输入不为零时停止仿真

4.8.3 连续系统模块(Continuous)

双击 Continuous 模块库，看到连续系统模块的种类如图 4-44 所示。表 4-7 介绍连续系统模块的功能。

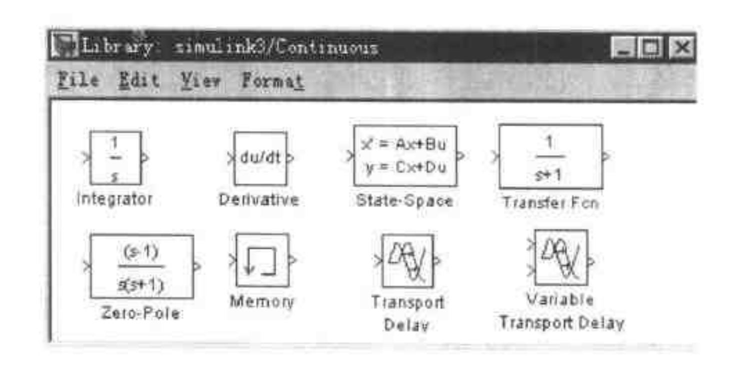


图 4-44 连续系统模块

表 4-7 连续系统模块简介

模 块 名	功 能 简 介	模 块 名	功 能 简 介
Integrator	积分环节	Zero-Pole	零极点模型
Derivative	微分环节	Memory	把第一步的输入延时后输出
State-Space	状态方程	Transport Delay	信号传输延时
Transfer Fcn	传递函数	Variable Transport Delay	按第二个输入指定的时间将第一个输入延时

4.8.4 离散系统模块(Discrete)

双击 Discrete 模块库，看到离散系统模块的种类如图 4-45 所示。表 4-8 介绍离散系统模块的功能。

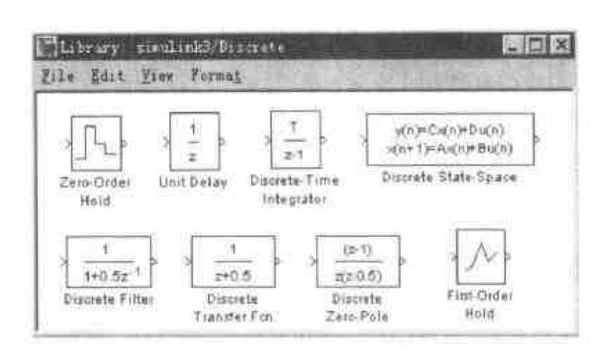


图 4-45 离散系统模块

表 4-8 离散系统模块简介

模 块 名	功 能 简 介	模 块 名	功 能 简 介
Zero-Order Hold	零阶保持器	Discrete Filter	离散滤波器（IIR、FIR）
Unit Delay	采样保持，延时一个周期	Discrete Transfer Fcn	离散传递函数
Discrete-Time Integrator	离散时间积分	Discrete Zero-Pole	离散零极点模型
Discrete State-Space	离散状态方程	First-Order Hold	一阶保持器

4.8.5 信号与系统模块 (Signals & Systems)

双击 Signals & Systems 模块库，看到信号与系统模块的种类如图 4-46 所示。

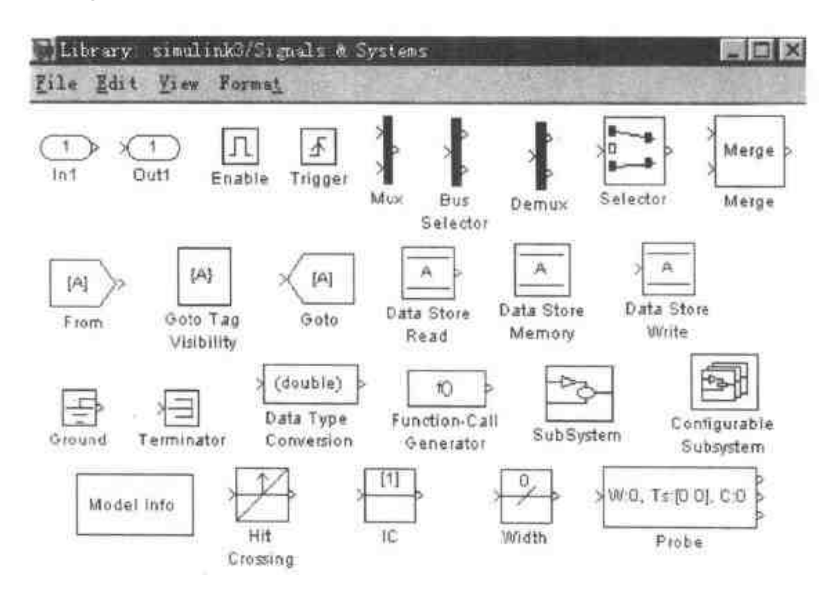


图 4-46 信号与系统模块

表 4-9 介绍信号与系统模块的功能。

表 4-9 信号与系统模块简介

模 块 名	功 能 简 介	模 块 名	功 能 简 介
In1，Out1	为子系统或模型提供输入、输出端口	Bus Selector	从输入中选择信号
Enable	放到子系统中建立使能子系统	Selector	选择输入的元素
Trigger	放到子系统中建立触发子系统	Merge	把输入信号合并为输出信号
Mux	把向量或标量组合成大的向量	From	接收标记(Tag)相同的 Goto 模块信号
Demux	把向量分成标量或小的向量	Goto Tag Visibility	定义 Goto 模块标记(Tag)的有效范围

续表

模　块　名	功　能　简　介	模　块　名	功　能　简　介
Goto	把信号输送到标记(Tag)相同的 From 模块	SubSystem	空的子系统
Data Store Read	从指定的数据存储器读数据	Configurable SubSystem	表示从用户指定的模块库里选择的任何模块
Data Store Memory	为数据存储器定义内存区域	Model　Info	显示模型的修改信息
Date Store Write	写数据到指定的数据存储器	Hit Crossing	检测输入信号的零交叉点
Ground	给未连接的输入端接地，输出 0	IC	设置信号的初始值
Terminator	连到未连接的输入端，终止输出信号	Width	检查输入信号的宽度
Data Type Conversion	数据类型转换	Probe	检测连线的宽度、采样时间和复数信号标记
Function-Call Generator	函数调用发生器		

4.8.6　数学运算模块(Math)

双击 Math 模块库，看到数学运算模块的种类如图 4-47 所示。表 4-10 介绍数学运算模块的功能。

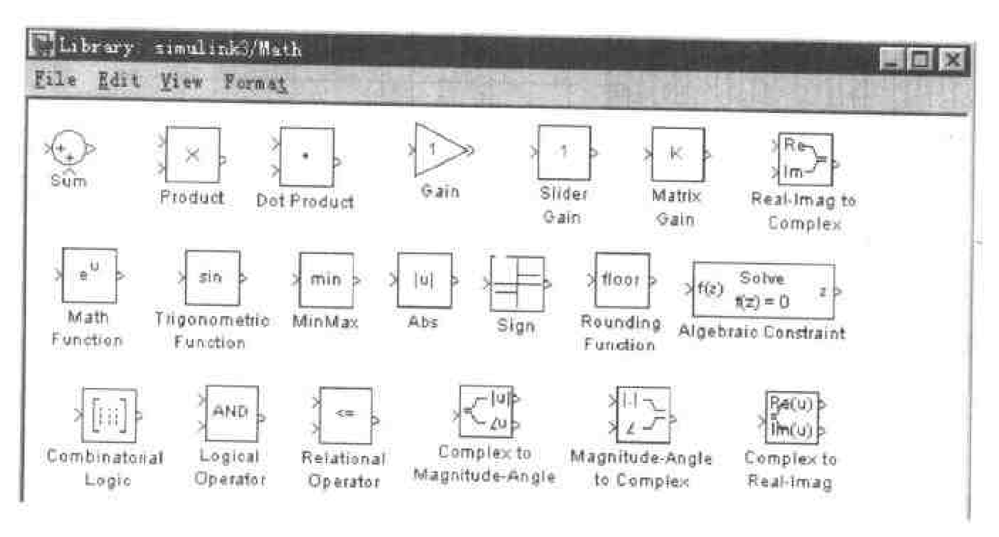


图 4-47　数学运算模块

表 4-10　数学运算模块简介

模　块　名	功　能　简　介	模　块　名	功　能　简　介
Sum	对输入求代数和	Abs	求绝对值或复数的模
Product	对输入求积或商	Sign	符号函数
Dot　Product	点积（内积）	Rounding Function	取整函数
Gain	常量增益（输入乘一个常数）	Combinatorial Logic	逻辑真值表
Slider Gain	可以用滑动条来改变的增益	Logical Operator	逻辑运算
Matrix	矩阵增益（输入乘以一个矩阵）	Relational　Operator	关系运算
Math Function	数学运算函数	Complex to Magnitude-Angle	求复数的模和辐角
Trigonometric Function	三角函数	Real-Image to Complex	根据模和辐角得到复数
Min Max	求最大值	Algebraic　Constraint	强制输入信号为零

下面以编码器的设计为例说明 Simulink 在数字电路仿真中的应用。

在数字电路中，编码器的应用极为广泛，例如键盘电路就应用了编码器。当我们敲击“a”键时，键盘送入计算机的只是一个脉冲值，而最后显示在屏幕上的却是 ASCII 码字符，这中间就是通过了一个编码器，该电路将键盘送入的脉冲信号转化成相应的八位 ASCII

码字符。

实现 8 线 3 线编码器，它的功能是对输入端的 8 个信号进行编码，输出三个二进制数。要求输入信号每次只有一个是 0，其余 7 个是 1。

它的功能是对输入端的 8 个信号进行编码，输出三位二进制数。要求输入信号每次只有一个是 0，其余 7 个是 1。用 Simulink 构成的编码器电路如图 4-48 所示。

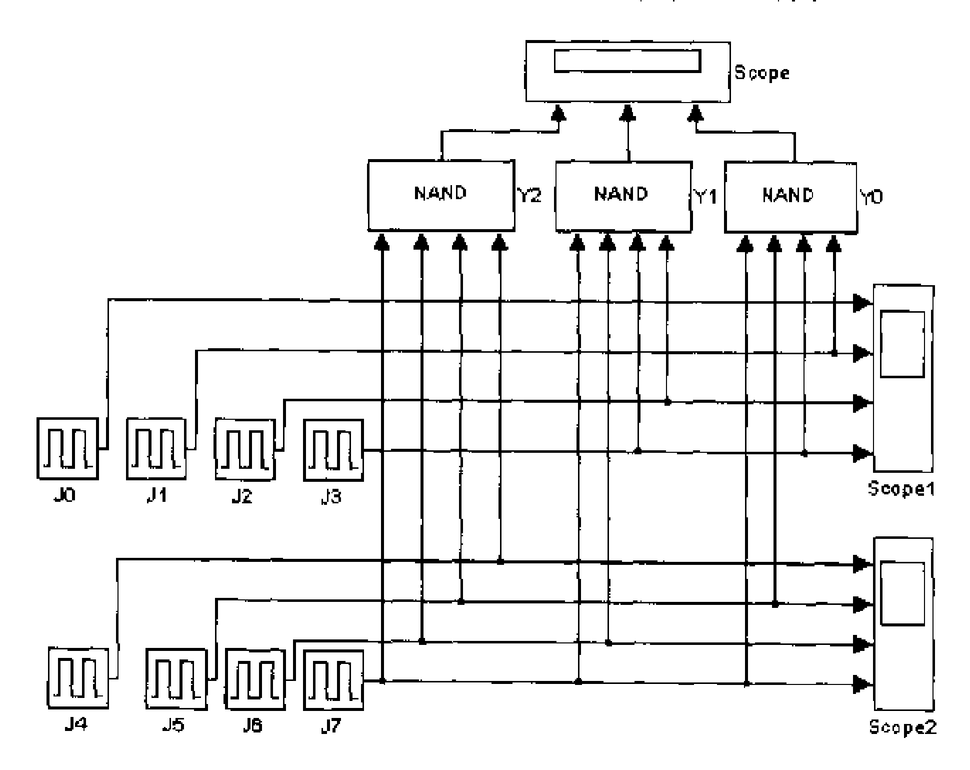


图 4-48 8 线 3 线编码器

在设计组合逻辑电路时，要先写出输入输出对应的真值表，根据 8 线 3 线编码器要实现的功能，可将其真值写出如表 4-11 所示。有了真值表，可以写出输入输出之间的逻辑函数表达式为：

$$Y_0 = \overline{J_1 \cdot J_3 \cdot J_5 \cdot J_7}$$
$$Y_1 = \overline{J_2 \cdot J_3 \cdot J_6 \cdot J_7}$$
$$Y_2 = \overline{J_4 \cdot J_5 \cdot J_6 \cdot J_7}$$

表 4-11 8 线 3 线编码器真值表

输入信号								输出信号		
J_0	J_1	J_2	J_3	J_4	J_5	J_6	J_7	Y_2	Y_1	Y_0
0	1	1	1	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	0	0	1
1	1	0	1	1	1	1	1	0	1	0
1	1	1	0	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1

用 Simulink 实现编码器这个数字电路方法如下。

（1）添加模块

运行 Simulink，打开模块浏览器，新建一个模型，将如下三种模块添加到模型中。依次打开 Simulink|Math|Logical Operator 添加与非门；依次打开 Simulink|Sources|Discrete Pulse Generator 添加离散信号激励源；依次打开 Simulink|Sinks|Scope 添加示波器，结果如图 4-49。

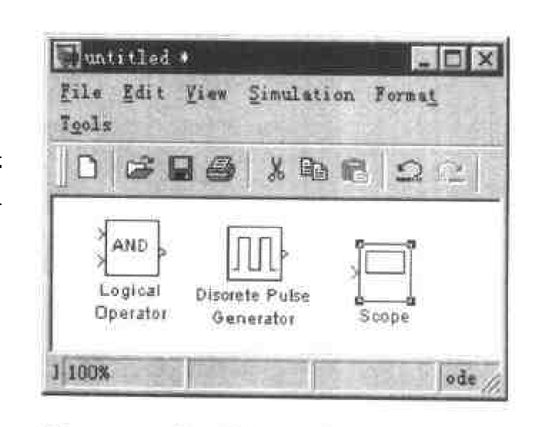


图 4-49　8 线 3 线编码器用到的三个模块

点击 Logical Operator 的名称，将会出现一个灰框，将鼠标放进去，将这个模块改为 Y0。接着点击选中与非门，不要松开鼠标，按住 Ctrl 键拖动这个模块到另一个位置，产生一个新的模块，是由 Y0 生成的，名字相应地改为 Y1。用同样办法生成 Y2。

用同样办法先将离散脉冲源的名字改为 J0，然后按住 Ctrl 键拖动 J0 七次，即可得到 8 个离散脉冲源，名字分别为 J0，J1，…，J7。将示波器也复制三个。这样所有的模块就都添加到模型中了。

（2）修改参数

双击模块 Y0，打开其属性对话框，如图 4-50 所示。将 Operator 修改为“NAND（与非）”，将 Number of input ports（输入节点数）修改为 4，然后点击 OK。其余两个逻辑运算模块 Y1、Y2 也做同样修改。

双击示波器模块 Scope，得到一个图形界面，在其工具栏上单击“ ”图标，可以打开示波器属性设置对话框，将 Number of axes（坐标轴数）改为 3，同样将示波器模块 Scope1、Scope2 的坐标轴数改为 4。

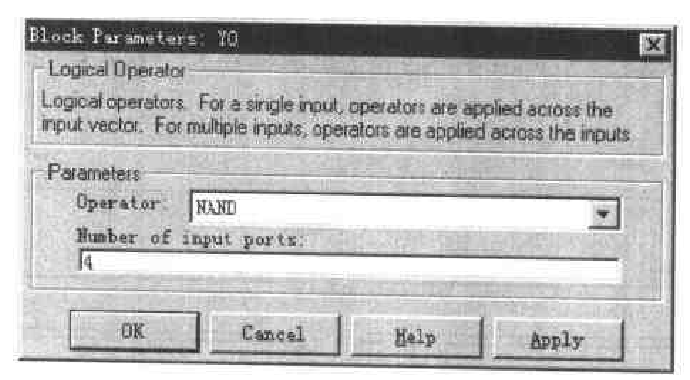


图 4-50　逻辑运算模块属性设置

双击离散脉冲源 J0，将看到它的属性对话框，如图 4-51 所示。可以修改对话框中的 5 个参数。5 个参数的含义如下。

① Amplitude：方波信号的幅度；

② Period：方波信号的周期（以采样时间为单位）；

③ Pulse width：脉冲宽度（即电平为 1 的时间，以采样时间为单位）；

④ Phase delay：相位延迟（即信号开始的时间，以采样时间为单位）；

⑤ Sample Time：采样时间长度，以秒为单位。

在这个例子中，需要 J0 到 J7 依次为低电平，所以将 J0 到 J7 的周期设置为 8，脉冲宽

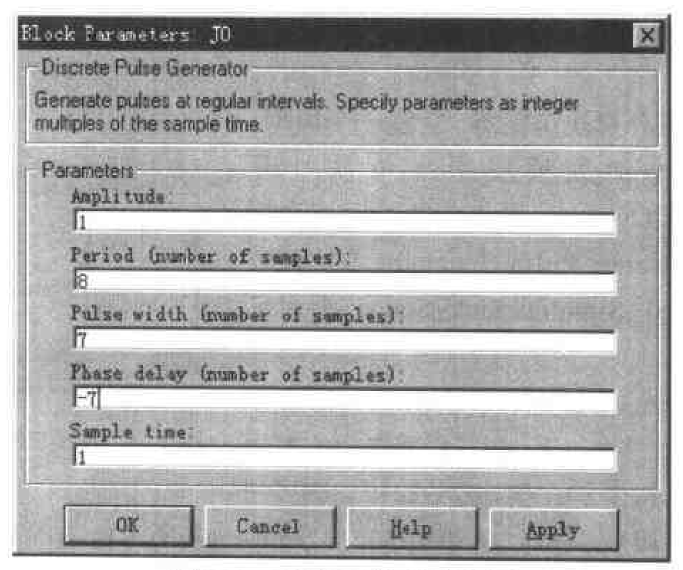


图 4-51　离散脉冲源参数设置

度设为 7，相位延迟依次为–7 到 0，幅度和采样时间采用默认值。这样在零时刻，J0 为低电平，其余输入为高电平；过一个采样时间，J1 变为低电平；到第七个采样时间，J7 变为低电平，实现了设计要求。

（3）连线

根据输入输出的逻辑表达式，将 J1、J3、J5、J7 接至 Y0 的输入，将 J2、J3、J6、J7 接至 Y1 的输入，将 J4、J5、J6、J7 接至 Y2 的输入，然后用示波器监视输入和输出，完成图 4-48 所示电路的连接。

（4）仿真

单击 Simulink|Parameters，将仿真时间设置为 0～20s，仿真步长和算法采用默认值。将这个模型存盘，命名为 bianma01。双击打开二个示波器，点击“■”图标开始仿真。

仿真结束后，在示波器 Scope1 和 Scope2 可观察到 8 个输入端的波形，如图 4-52 所示。从 J0 到 J7 以 8 为周期，依次为 0 电平。

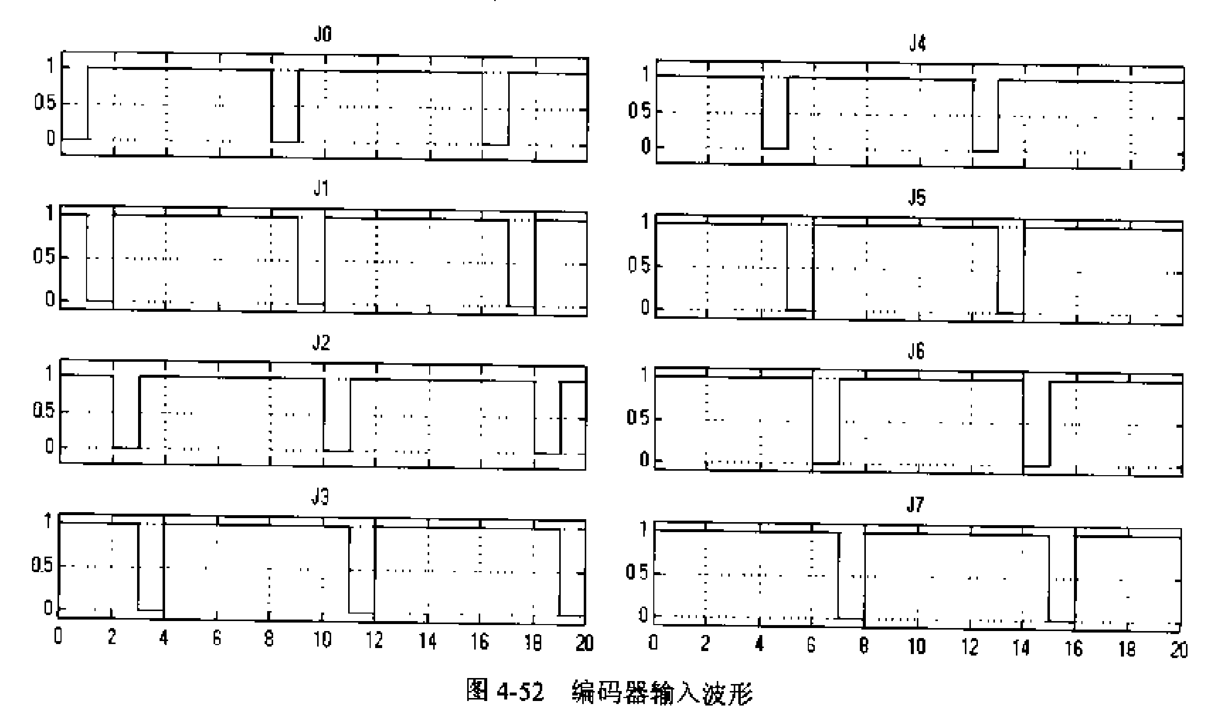


图 4-52　编码器输入波形

在示波器 Scope 可观察到 3 个输出端的波形，如图 4-53 所示。从中可以看到输出的三位二进制码依次为：000、001、010、011、100、101、110、111，实现了编码的功能。

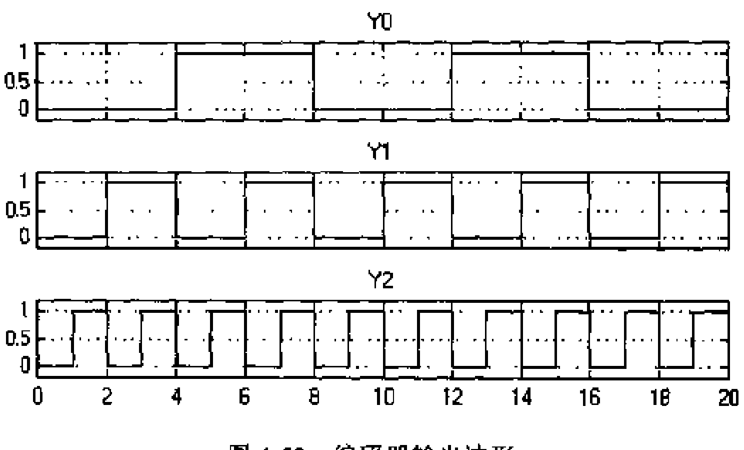


图 4-53 编码器输出波形